

摘要

随着物流行业的发展，物流企业的业务内容越来越繁杂，客户对服务质量的要求也越来越高，企业管理难度不断加大，容易出现企业内部运作流程混乱、人员变动频繁、作业质量参差不齐等问题，导致管理成本直线上升、客户服务水平难以提高。在这种情况下，物流企业急需借助一套科学、适用的标准化运作流程来摆脱困境。运用标准化的运作流程能使企业有效地化繁为简，使企业在经营模式的扩张中、在人员的变动更替中不变样、不走味，提高作业效率和服务水平，提高企业的快速响应速度，降低管理和运营的成本，为物流企业创造出更多利润，为客户提供更好的服务。

我们在对安得物流公司的案例背景进行认真研究并对安得物流公司的运作进行实地考察的基础上，设计了这套物流标准化运作流程方案，力求做到科学、简明、适用性强。

方案中以大量的标准化运作流程图、流程表以及必要的操作说明，从安得物流公司的主营业务来展开设计，然后运用所设计的标准化运作流程对案例中所提出的安得物流公司业务流程中所存在的问题提出解决方案，并根据物流敏捷调度优先技术模型编写出车辆调度程序以利于简化操作流程，最后我们探讨了物流标准化运作流程方案实施前效果分析和评价的问题，提出借助物流仿真软件进行方案效果分析和评价的方法，并辅以示例说明。方案不仅从思路和内容上都体现了创新，而且大量的标准化运作流程图、流程表以及方案的完整性更展示了这套设计方案的特色。我们希望这套设计方案对于安得物流公司以及同类型的第三方物流公司具有一定的实际应用价值与参考作用。

关键字：标准化 运作流程 仓储 配送 运输

目录

1 导言	1 -
2 物流标准化运作流程的总体设计思路	2 -
2.1 方案设计背景	2 -
2.2 物流标准化运作流程的设计思路	2 -
2.2.1 物流标准化运作流程设计的目标与原则	2 -
2.2.2 物流标准化运作流程设计的总体思路	2 -
2.3 物流运作总体流程	4 -
2.3.1 仓储管理	4 -
2.3.2 配送管理	4 -
2.3.3 运输管理	5 -
2.3.4 物流运作总体流程	6 -
3 仓储流程标准化设计	7 -
3.1 安得公司仓储现状的分析	7 -
3.2 仓储流程标准化设计概述	8 -
3.2.1 仓储流程标准化设计的目标	8 -
3.2.2 仓储流程标准化设计的基本内容	8 -
3.3 仓储流程标准化具体设计	8 -
3.3.1 入库管理的标准化	9 -
3.3.2 在库管理的标准化	15 -
3.3.3 出库管理的标准化	18 -
3.4 安得公司仓储问题解决方案	19 -
3.4.1 对“仓储设计”问题提出的解决方案	19 -
3.4.2 对“主动持续补货”问题提出的解决方案	21 -
3.4.3 对“花城分公司”问题提出的解决方案	23 -
4 配送流程标准化设计	26 -
4.1 安得公司配送现状的分析	26 -
4.2 配送流程标准化设计概述	27 -
4.2.1 配送流程标准化设计的目标	27 -
4.2.2 配送流程标准化设计的基本内容	27 -
4.3 配送流程标准化的具体设计	27 -
4.3.1 订单处理流程的标准化	28 -
4.3.2 备货流程的标准化	34 -
4.3.3 分拣及配货流程的标准化	35 -
4.3.4 配装流程的标准化	37 -

4.3.5 送货及结算流程的标准化.....	- 40 -
4.4 安得公司配送问题解决方案.....	- 42 -
4.4.1 对“P 分公司共同配送”问题提出的解决方案.....	- 42 -
4.4.2 对“配送增值服务”问题提出的解决方案.....	- 45 -
4.4.3 对“淡旺季作业不均”问题提出的解决方案.....	- 47 -
5 运输流程标准化设计	- 50 -
5.1 安得公司运输现状分析.....	- 50 -
5.2 运输流程标准化设计概述.....	- 51 -
5.2.1 运输流程标准化设计的目标.....	- 51 -
5.2.2 运输流程标准化设计的基本内容.....	- 51 -
5.3 运输流程标准化具体设计.....	- 51 -
5.3.1 受理托运和订立运输合同流程的标准化.....	- 52 -
5.3.2 运输方式选择的标准化.....	- 54 -
5.3.3 车辆调度流程的标准化.....	- 59 -
5.3.4 车辆配装流程的标准化.....	- 64 -
5.3.5 在途监控流程的标准化.....	- 65 -
5.3.6 送达服务交割与结算流程的标准化.....	- 67 -
5.4 安得公司运输问题解决方案.....	- 68 -
5.4.1 对“铜川 J 客户盈利率低”问题提出的解决方案.....	- 68 -
5.4.2 对“要么扭亏，要么下课”问题提出的解决方案.....	- 71 -
5.4.3 对“呼叫中心”问题提出的解决方案.....	- 74 -
6 物流标准化运作流程方案的实施效果评价	- 77 -
6.1 评价的目标.....	- 77 -
6.2 评价的指标.....	- 77 -
6.3 评价的方法.....	- 78 -
7 结束语	- 81 -
参考文献	- 82 -
附录	- 83 -
附录一 物流敏捷调动优先技术的假设和计算.....	- 83 -
附录二 物流敏捷调动优先技术在 Visual C#环境下编程和操作说明.....	- 89 -
附录三 物流敏捷调动优先技术在 Visual C#环境下的应用示例.....	- 113 -

1 导言

运作流程标准化是企业化繁为简、提高管理和运营效率的有效工具，它针对经营管理中的每一个环节、每一个部门、每一个岗位，以人为本，制定细致化、科学化、数量化的标准并严格按照标准实施管理，极大地提高了工作效率，使企业能够以较少的投入获得较大的产出。

安得物流有限公司已在全国拥有 100 多个网点，在 50 多个大中城市设立了 60 多万平方米的仓库，能为客户提供 7×24、管理一体化的仓储服务；强力整合公路、铁路、航空、水运的运输资源，实现多种运输模式最佳组合；“精益配送”在全国直辖市、省会城市以及经济较发达地区均已具有一定规模。发展到这一步，安得公司更要加强管理，实现运作流程标准化，才能更好地与国际接轨。

在实现物流运作流程标准化之前，人员变动是物流企业最难以适应的情况之一。安得公司也遇到这种情况，在旺季作业时，新职员难以胜任分配的任务；货物配装时随个人经验而进行；订单录入因个人习惯而造成了内容的遗漏……由于欠缺标准化的管理，令物流企业运作失去稳定。运作流程标准化能够较好解决这些问题，即使是一位新员工，只要严格按照标准去做，也能保证工作顺利展开。

以上种种情况，都说明了安得公司实施运作流程标准化的必要性。因此，在对安得公司的案例背景进行认真研究并对安得公司的运作进行实地考察的基础上，我们设计了这套物流运作流程标准化方案。方案包含大量的标准化运作流程图和流程表以及必要的操作说明，层次清晰，结构完整。自始至终我们注重所设计的物流标准化运作流程的可行性和可操作性。方案不仅从思路和内容上都体现了创新，而且大量的标准化运作流程图、流程表以及方案的完整性更展示了这套设计方案的特色。

本方案首先以安得公司的业务作为背景提出物流标准化运作流程的总体设计思路，根据我们的流程层级图简要概述第三方物流企业主营业务运作的总流程及各主要运作环节的流程。接下来根据物流标准化运作流程的总体设计思路分别从仓储、配送、运输三个方面详细设计具体的标准化运作流程和操作说明，层层剖析，在总流程的基础上深入重点。

仓储方面，主要设计的标准化运作流程有：入库管理中的保管场所安排、入库前准备、装卸搬运；在库管理中的盘点、异状处理、补货请求；出库作业中的出库前准备、备货、理货。配送方面，主要设计的标准化运作流程有订单处理流程的标准化、分拣及配货流程的标准化、分拣及配货流程的标准化、配装流程的标准化、送货及结算流程的标准化。运输方面，主要设计的标准化运作流程有：受理托运和订立运输合同流程的标准化、运输方式选择流程的标准化、车辆调度流程的标准化、车辆配装流程的标准化、在途监控流程的标准化、送达服务交割与结算流程的标准化。

然后，运用所设计的标准化运作流程对案例中所提出的安得物流公司业务流程中所存在的问题提出解决方案，并根据物流敏捷调度优先技术模型编写出车辆调度程序以利于简化操作流程，从而减少安得物流公司运作流程上由于人为因素导致的问题与风险，实现流程运作的标准化。最后我们探讨了物流标准化运作流程方案实施前效果分析和评价的问题，提出借助物流仿真软件进行方案效果分析和评价的方法，并辅以示例说明。

2 物流标准化运作流程的总体设计思路

2.1 方案设计背景

安得公司是专业化、规模化的第三方物流公司，其业务涉及仓储业、货物运输业务、为客户提供货物储存、保管、中转等仓储服务；涵盖全国直辖市、省会城市以及经济较发达地区的“精益配送”；货物快递服务和运输代理服务等等。但是不断变化的物流环境对安得公司提出了更高的要求。随着经济的发展和客户需求增加，安得物流公司正考虑一个转型的问题，因此正逐步在开拓一些新的业务，例如第四方物流公司的咨询业务等，安得公司利用自身的信息系统可以为客户提供专业的物流方案的设计。

一直以来，安得公司都立志于建立一流的物流服务品牌 and 专业的物流服务平台。要实现这个目标，安得公司必须做到以下几点：在业务，做到体制上与国际接轨；在流程上，做到流程简明，物畅其流；在管理上，做到权责分明；在财务上，做到在已有的基础上运用最少的资金投入得到最大的收益，而要实现以上目标都要求安得公司做到——物流运作流程的标准化！

2.2 物流标准化运作流程的设计思路

2.2.1 物流标准化运作流程设计的目标与原则

目标：为安得公司制订一套科学、简明、适用性强的物流标准化运作流程，处理纷繁复杂的业务，使整个公司的运作流程有章可循，降低管理和运营的成本，提高运作效率，创造出更多利润、提高服务水平。

原则：科学、简明、适用性强。

2.2.2 物流标准化运作流程设计的总体思路

我们紧扣目标，在对安得物流有限公司的案例背景进行认真研究和实地考察的基础上，加上合理的假设来展开设计。我们设计方案的总体思路主要是坚持以物流运作流程的标准化为主线，从宏观的初级流程到一级流程，从一级流程扩充到二级流程，再对二级流程里的作业环节（即三级流程）细化，层层深入，环环紧扣，最终产生精细、标准、合理的作业流程。具体设计如图 2-1 所示：

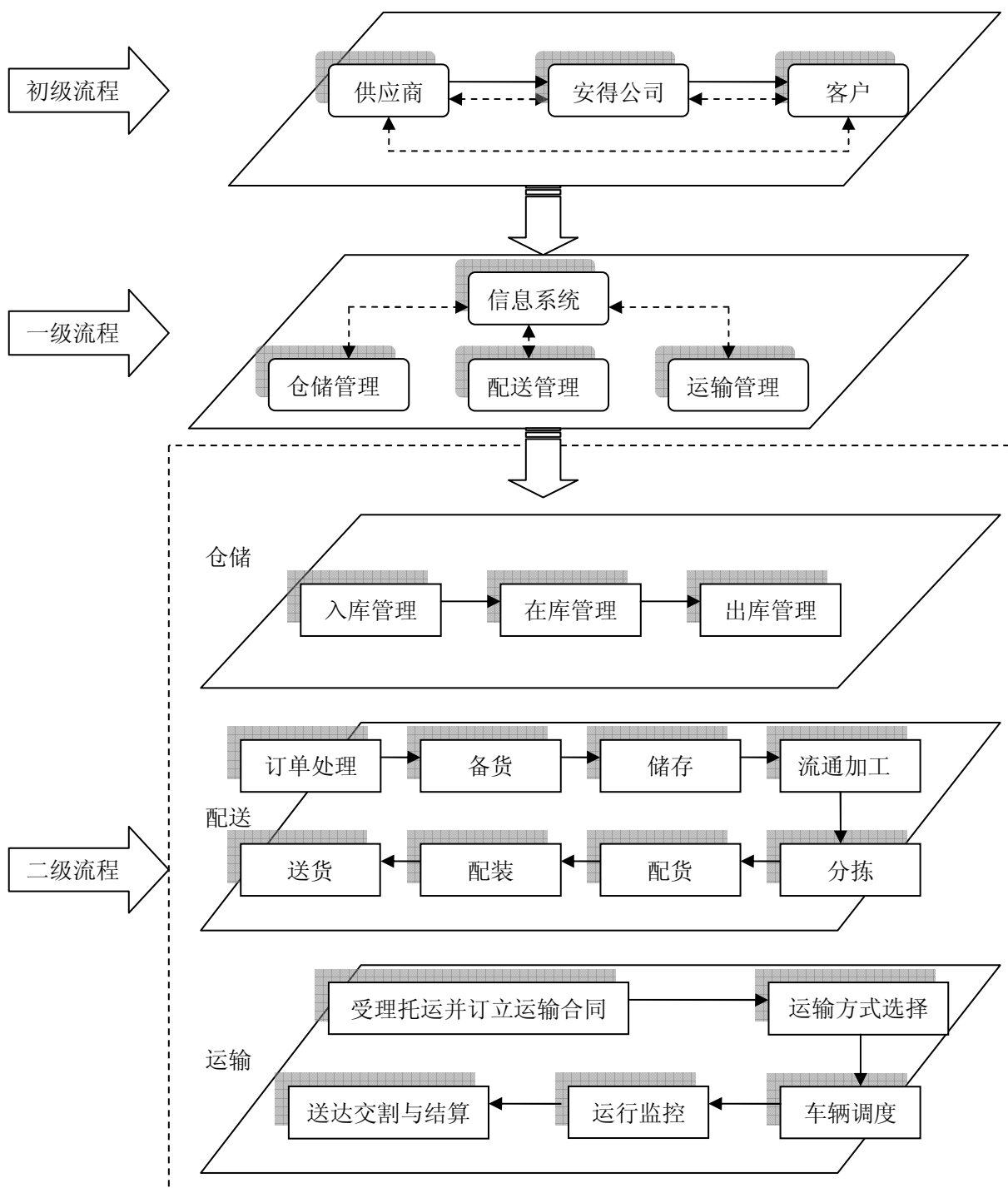


图 2-1 流程层级图

步骤一：设计了初级的物流标准化运作流程，理清安得公司与合作伙伴之间的关系。在这一层的设计中，我们更侧重于宏观方面，得出整体思路。

步骤二：依据初级流程图，设计安得公司内部运作的总体流程图即一级流程图。在这级的设计中，信息系统贯穿了整个流程的核心。

步骤三：分别从仓储、配送、运输三个方面展开设计二级流程图，使流程细化，方便我们分析研究安得公司的问题主要集中在哪些流程上，以确定设计的重点，逐个击破。

步骤四：深挖在步骤三确定下来的设计重点，最后集中于具体的标准化运作流程和操作说明。如仓储方面的类似于可承诺库存（ATP）的主动持续补货；配送方面的订单分类、输入订单资料和 P 分公司的共同配送；运输方面的车辆调度和越库运输。

仓储、配送、运输三部分运作流程标准化设计完成后，我们再利用所设计的方案来解决安得案例中出现的若干问题。

2.3 物流运作总体流程

现阶段安得公司的主要物流运作还是集中在仓储管理、配送管理、运输管理三方面，如图2-2所示：

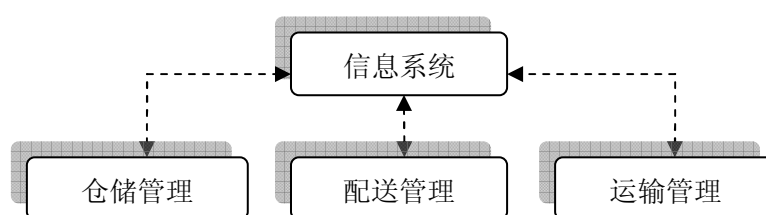


图 2-2 主要运作业务简图

2.3.1 仓储管理

如上面思路图所示，仓储作业流程主要包括入库管理、在库管理、出库管理，它的下一级的作业流程如表 2-1 所示：

表 2-1 仓储管理运作流程表

仓储管理		
入库管理	在库管理	出库管理
(1) 入库前准备 (2) 验收 (3) 决定储存单位 (4) 装卸搬运 (5) 苫垫堆码 (6) 办理入库手续	(1) 在库保管 (2) 盘点 (3) 异状处理 (4) 补货请求	(1) 出库前准备 (2) 核对出库凭证 (3) 备货 (4) 理货 (5) 全面复核查对 (6) 登账 (7) 交接清点

2.3.2 配送管理

如图，配送作业流程包括订单处理、备货、储存、流通加工、分拣、配货、配装、送货，其下一级的作业流程如表 2-2 所示：

表 2-2 配送管理运作流程表

配送管理			
订单处理	备货	储存	流通加工
(1) 接受订货 (2) 订单确认 (3) 设定订单号码 (4) 建立客户档案 (5) 存货查询及按订单分配存货 (6) 计算拣取的时间 (7) 依订单排定出货时间和拣货顺序 (8) 分配后存货不足的处理 (9) 订单处理和资料输出	(1) 制订进货计划 (2) 商品送达 (3) 卸货 (4) 收货 (5) 货品的编号标示 (6) 货物分类 (7) 核对单据和信息 (8) 货品验收 (9) 处理进货信息	(1) 临时货物的存放 (2) 分拣区货物存放 (3) 安全库存 (4) 分类存放	(1) 包装 (2) 贴标签 (3) 分类 (4) 组装
分拣	配货	配装	送货
(1) 形成拣货资料 (2) 选择拣货方法 (3) 选择拣货路径 (4) 搬运或行走 (5) 拣取	(1) 分货 (2) 配货检查 (3) 包装、打捆	(1) 拼箱 (2) 拼装	(1) 车辆调度 (2) 车辆配装 (3) 运送 (4) 送达服务与交割 (5) 费用结算

2.3.3 运输管理

运输作业流程包括受理托运并订立运输合同、车辆调度、运行监控、送达交割与结算、退货处理，其下一级的作业流程如表 2-3 所示：

表 2-3 运输管理流程表

运输管理				
订立运输合同	运输方式选择	车辆调度	运行监控	送达交割与结算
(1) 受理装车和编制单证 (2) 订立运输合同 (3) 合同的变更与解除	(1) 自营运输 (2) 协同运输 (3) 运输外包	(1) 车辆调度 (2) 货物装配	(1) 在途监控 (2) 呼叫中心	(1) 送达服务与交割 (2) 结算

2.3.4 物流运作总体流程

综合仓储、配送和运输三个方面，可以得出物流运作总体流程如图 2-3 所示：

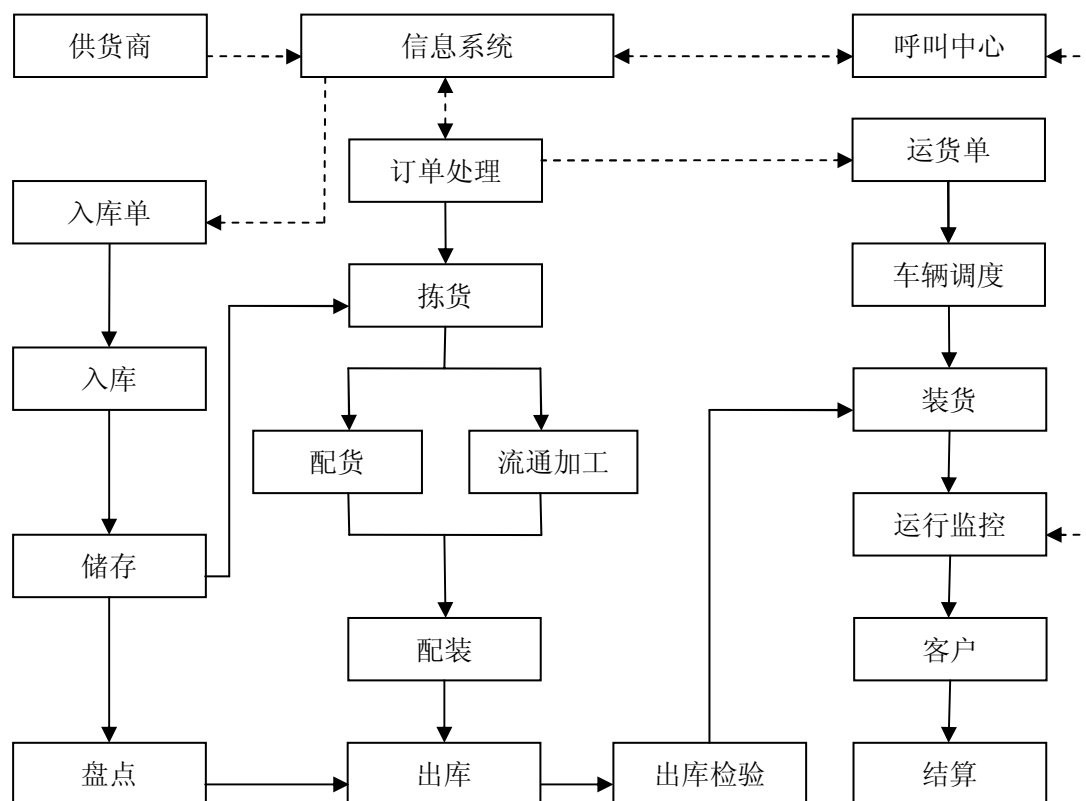


图 2-3 物流运作总体流程图

下面我们将在第三、第四、第五部分分别来详细说明仓储、配送和运输方面的标准化运作流程和操作说明。

3 仓储流程标准化设计

3.1 安得公司仓储现状的分析

目前安得公司根据业务规模及其范围在全国省会城市及部分二、三级城市建立了 50 多个、总面积达 60 多万平方米的仓储网点。可提供普通货品仓储服务和银行金融监管仓储服务，零配件仓储服务，并能提供库存库龄分析、条码管理、退换货管理等增值服务。公司掌握了各地大量的仓储资源，能满足客户变化着的需求，实行 7×24 小时全天候物流服务，仓储管理的全面信息化能使客户即时查询货品的信息。

当然，安得公司的仓储也会存在许多的问题。比如案例中提出的装卸效率低，出现排队、磨洋工等现象。我们用鱼骨分析法来分析仓储运作中出现的问题，如图 3-1 所示：

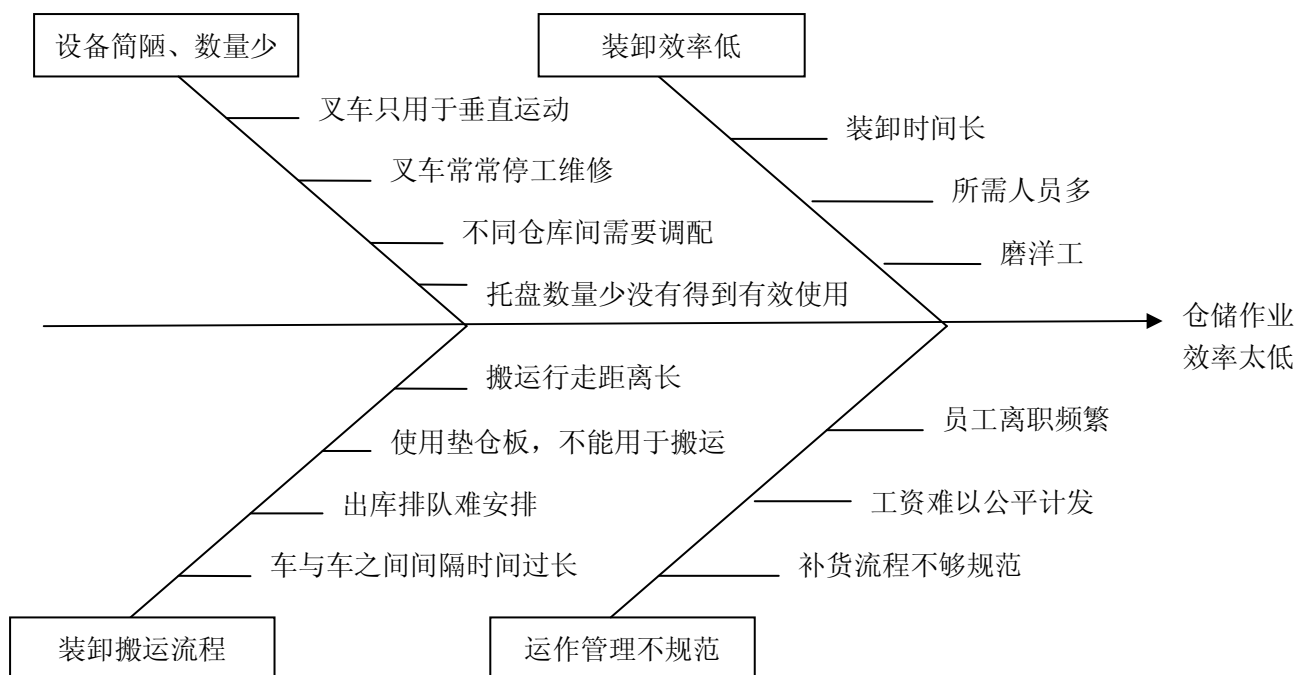


图 3-1 仓储问题鱼骨分析图

从鱼骨分析图，我们得出了四方面的原因：

1. 设备简陋，数量少。只有 5 台柴油叉车及 14 台液压手推车。仓库使用垫仓板，缺少托盘、立体货架等设备。柴油叉车使用范围过小，没能使装卸效率有效提高，还需要常常停工修理。人员、设备有时会在不同仓库间调来调去，麻烦而且增加了工作量。

2. 工作效率低，每车需要 6—7 装卸工人，同时只能装卸 8—9 车，而且时间长，平均装完一车要 70 分钟，卸完一车要 60 分钟。

3. 操作流程不顺畅，卸完一车到下一车之间等待时间过长，车辆对位、临时的货位查找、人员器具组织需要花费许多时间，而这些是可以考虑事先安排的。

流水作业经常出现交替等待。出库作业排队现场严重，不同仓库几次排队，而且排队先后顺序难令人满意。使用垫仓每次装卸车要多经过转板、上板等环节。单次出入库行走距离 80 米，更延长了时间，也是导致交叉等待的原因之一。

4. 运作管理不规范。某些流程（如补货）的做法不够规范。另外，由于缺乏合理的绩效考核，没有较好的激励机制，无法提升员工的忠诚度，有些分公司出现了员工频繁离职的情况。

以上的问题有许多是由于仓储运作流程不合理而造成的，随着公司发展，情况日益严重，安得公司急需借助一套有效的仓储管理体系管理，管理仓储中的各项运营活动。更重要的是把这套体系制订为标准，让员工对照标准进行自我管理，清楚自己的任务和目标，管理人员也很容易对其进行考核。

3.2 仓储流程标准化设计概述

3.2.1 仓储流程标准化设计的目标

仓储的运作流程标准化是仓储管理最有效的工具之一，尤其是在安得公司现有人员设备条件的限制下，通过实行仓储运作流程的标准化，能够有效的实行科学管理，能保证仓储作业流畅，提高整体的工作效率，以达到资源利用的最大化，进而解决安得公司人力缺乏、设备简陋、出库作业排队现场严重等带来的问题。

3.2.2 仓储流程标准化设计的基本内容

根据安得公司的仓储问题鱼骨分析图，我们总结出仓储的运作流程中需要注意的环节主要包括：

1. 入库管理中的保管场所安排、入库前准备、装卸搬运；
2. 在库管理中的盘点、异状处理、补货请求；
3. 出库作业中的出库前准备、备货、理货。

针对仓储流程中的这些环节，我们制订标准化的运作，并结合案例对这些运作流程的标准化进行具体说明，其中还包括在运作流程中员工该遵守哪些原则，遇到问题怎样解决。希望通过实行仓储运作流程标准化能使安得公司的仓储运作管理规范化、程序化和科学化，从而提高工作效率。

3.3 仓储流程标准化具体设计

仓储管理总体流程图如下：

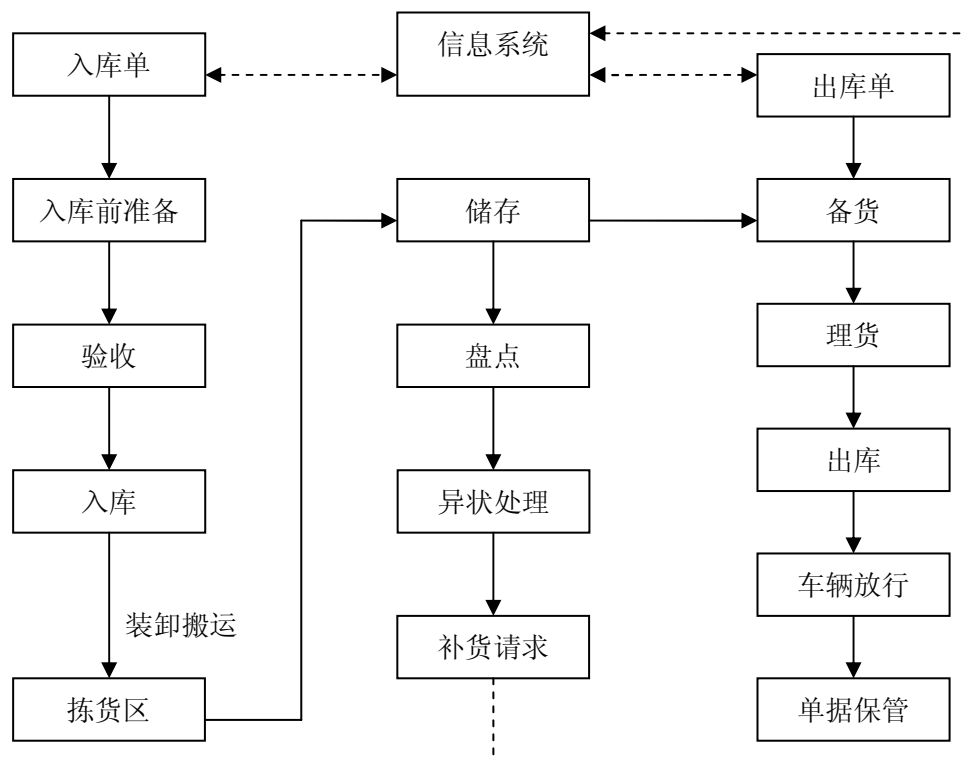


图 3-2 仓储管理流程图

下面，我们根据以上的仓储管理总体流程图，把仓储管理的流程分为入库管理、在库管理、出库管理三大程序，并针对其中的重要环节进行深入分析。

3.3.1 入库管理的标准化

入库是仓储工作的第一步，它的工作内容主要包括货物的入库准备、办理入库手续、货物的检验、理货以及对搬运装卸合理化的管理。

入库管理标准化以后的流程如图 3-3 所示：

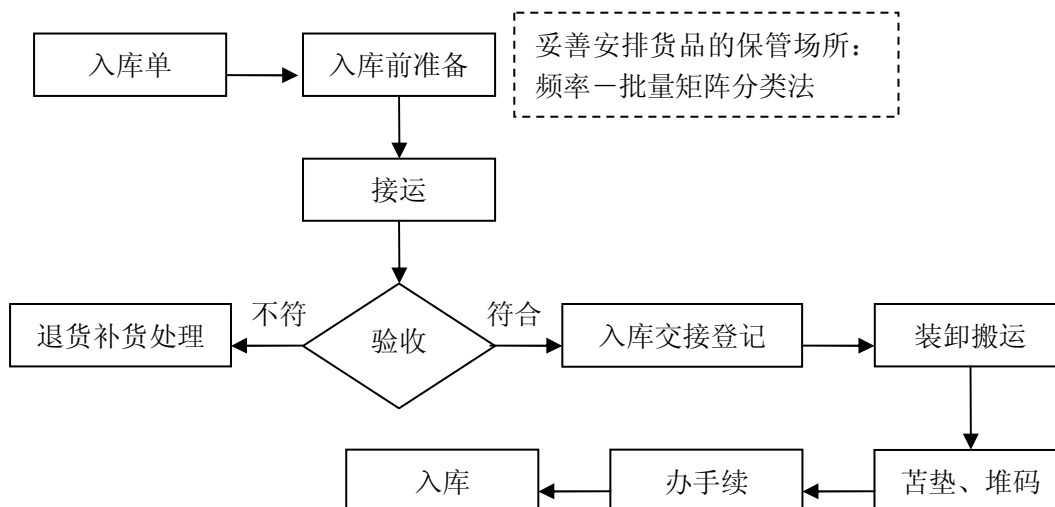


图 3-3 入库管理流程图

接下来，我们对入库管理流程中的入库前准备、验收、装卸搬运的具体运作设立标准化。

（一）入库前准备作业的标准化

入库前准备是仓储作业流程的第一步，入库前准备是否充足对后期作业有重要的影响。而产品入库前，首先又要考虑产品的储存安排，如安得公司的 Y 分公司的仓库内的搬运距离长等问题，合理的保管场所规划能使之得到有效解决。

1. 安排货品的保管场所的标准化操作，如图 3-4 所示：

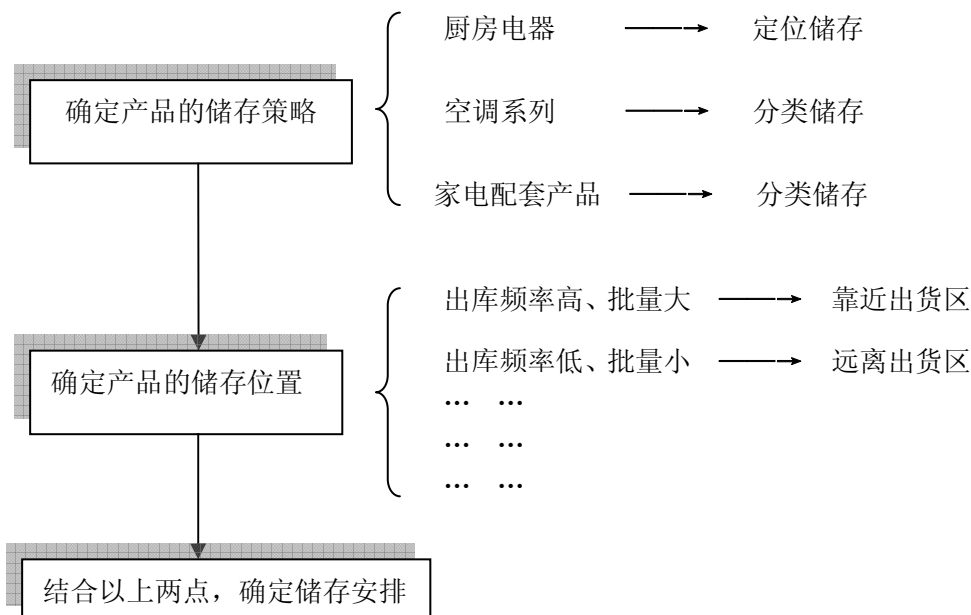


图 3-4 储存安排流程图

（1）确定产品的储存策略。

产品的储存策略包括定位储放、随机储放、分类储放和共同储放。

其中定位储存容易管理，而且货品的储位可按周转率大小（畅销程度）安排，以缩短出入库搬运距离。但却需要较多的储存空间。因此适合于厂房空间大，多品种、小批量重要物品。

假设 T 客户的主要产品分为空调、厨房电器（电饭煲、冰箱、微波炉等）、家电配套产品（压缩机、电机、磁控管等），按以下方式选择：

- ① 厨房电器的周转率高，品种多，单一种类的批量较小，而安得给予 T 客户的仓库面积有 35000 或 42000 平方米，厂房空间较大，因此，厨房电器应该选择定位储存；
- ② 空调系列与厨房电器相比种类较少、体积大，进出库高峰期集中在夏季，因此，选择分类储放；
- ③ 家电配套产品相关性、流动性、尺寸、特征明显，选择分类储放。

（2）确定产品的储存位置。

安得公司的仓管员根据订单的定货量，应用频率-批量矩阵进行评分，从而确定货物的储存位置，选择总体上最适合库内货物进出要求的储存方式。划分货物不同的出入库频率和不同的出入库的批量，对于出入库频繁而且出入库批量大的放在最便于出入库的位置，保证既做到抓住重点又照顾一般。而 T 客户的货物大部分为小家电，如电饭煲、电热杯等，较为轻巧，因此可以忽略重量、体积因素。如表 3-1 所示：

表 3-1 频率—批量矩阵

频率 \ 批量	I	II	III	IV
A	1	3	6	10
B	2	4	7	13
C	5	8	11	15
D	9	12	14	16

在上表中，

A 类：出入库频率最高的产品

B 类：出入库频率较高的产品

C 类：出入库频率较低的产品

D 类：出入库频率最低的产品

I 类：出入库批量最大的产品

II 类：出入库批量较大的产品

III 类：出入库批量较小的产品

IV 类：出入库批量最小的产品

综合考虑出入库频率以及出入库批量，对各类产品进行打分，如 1 分代表出入库频率和出入库量都是最大的商品，因此，应该安排在进出库最方便的位置，如此类推。安得可以根据情况变动更改表中的分值。

由于各种商品的自然属性、包装、尺寸等不同，对保管条件的要求也不同，因此在储存安排时应该加以考虑。

根据频率—批量矩阵，可拟定储存场所安排如表 3-2 所示：

表 3-2 储存场所安排表

产品摆放优先级	摆放位置
1—5	靠近进出口，大储区/中储区
6—10	较近出口，中储区
11—15	离出口较远，中小储区
16—20	较为偏远的角落

(3) 结合以上两点，确定储存安排。比如电饭煲、电视机、空调三类产品保管场所的安排：

① 厨房电器的保管场所选择，如图 3-4 所示：

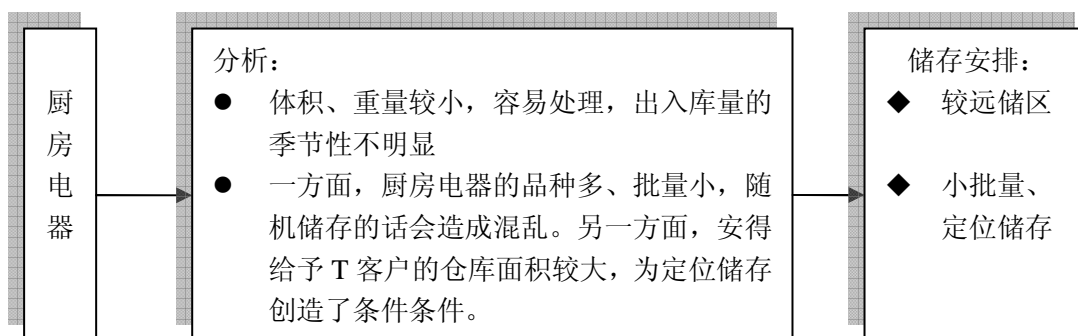


图 3-5 厨房电器保管场所图

即 T 客户的厨房电器应该选择小批量的存放在较远的固定储位上。

② 空调与电视机系列分析

a) 仓库选择

假设电视机、空调是 T 客户的主要进出库产品，两者出入量大、季节性明显，因此，我们选择放置在不同的仓库内，以避免两者都需要靠近进出口的储位而引起冲突。同时， $42000 - 35000 = 7000$ 平方米，即下半年的仓库需求量比上半年的仓库需求量多出 7000 平方米，我们可以假设这 7000 很大部分与电视机有关，因此，电视机选择放在 4 仓库（24000 平方米），空调选择放在 3 仓库（19000 平方米）。

b) 库内保管场所选择，如图 3-6 和图 3-7 所示：

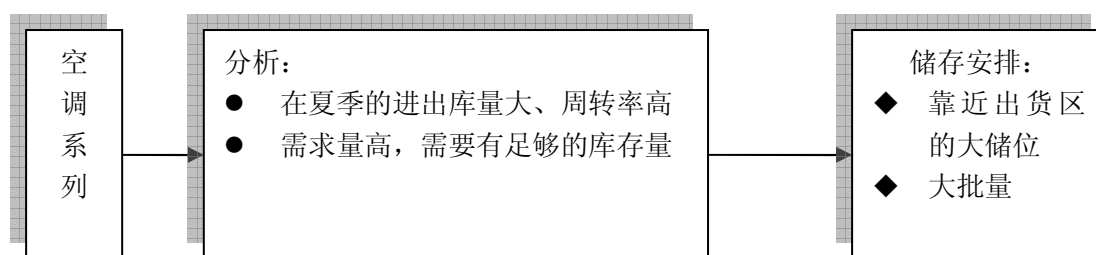


图 3-6 空调系列保管场所图

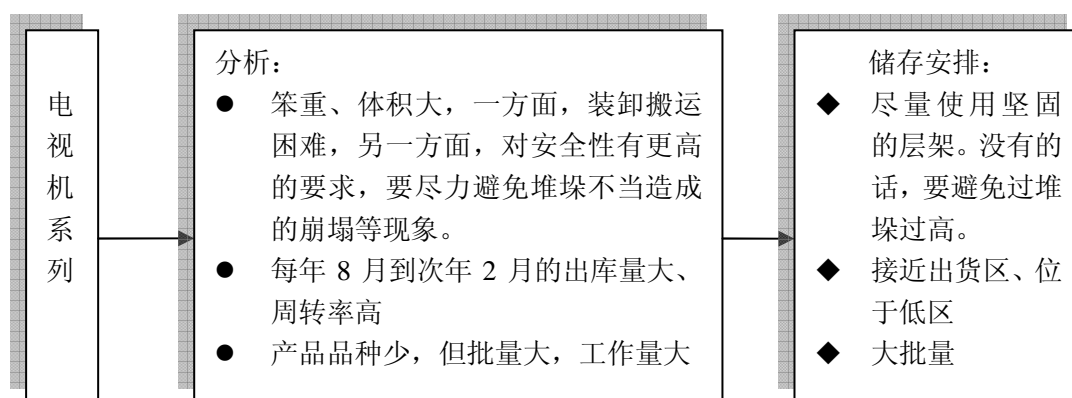


图 3-7 电视机系列保管场所图

即 T 客户的空调应该选择靠近出货区的大储位，而电视机一方面尽量使用托盘货架和叉车作业，另一方面，要选择接近出库区的大储位。值得注意的是，两者都有明显的季节性，因此所占据的靠近出货区的储位比例要适当，以免影响其他产品的进出库。

2. 每次入库前充分需要准备以下内容，如表 3-3 所示：

表 3-3 入库前准备内容

准备内容	注意要点
熟悉入库货物	品种、规格、数量、包装状态、体积、到库确切时间、存放期、保管要求等
掌握仓库库场情况	库容、设备、人员的变动情况

制订仓储计划	货物情况、仓库情况、设备情况
合理组织人力	安排好货物验收人员、搬运堆码人员以及货物入库工作流程
做好货位准备	清洁货位、检查照明通风设备
准备苫垫材料、作业用具	苫垫方案、材料
验收准备（方法，点数、称量、测试工具）	确定验收方法，准备各项工作的工具
装卸搬运工艺设定	根据货物、货位、设备条件、人员制订卸车搬运工艺
文件单证准备	入库记录、理货检验单、料卡、残损单等

（二）验收流程的标准化

产品接运入库后，需要对产品进行验收，以保证产品的数量、质量。

验收标准化后的主要流程如表 3-4 所示：

表 3-4 验收流程说明表

流程	执行要点
● 验收准备	<ul style="list-style-type: none"> ◆ 收集、整理并熟悉各项验收凭证、资料及有关验收要求。 ◆ 准备所需的计量器具、检测仪器仪表等。 ◆ 落实入库货物的存放地点，选择合理的堆码垛型和保管方法。 ◆ 准备所需的苫垫堆码物料、装卸机械、操作器具和负责验收作业的人力。 ◆ 要求质量检验时，要预先通知检验部门到库检验。
● 核对验收单证	<ul style="list-style-type: none"> ◆ 审核验收依据——入库通知单、订货合同、协议书等。 ◆ 核对供货单位提供的验收凭证——质量保证书、装箱单、磅码单、说明书、保修卡、合格证等。 ◆ 核对承运单位提供的运输单证——提货通知单、货损记录、公路运输交接单。
● 确定抽检比例	<ul style="list-style-type: none"> ◆ 商品价值——价值高，则抽检比例大。 ◆ 商品的性质——不稳定，则抽检比例大。 ◆ 气候条件——怕潮怕冻，则抽检比例大。 ◆ 运输方式和运输工具——运输影响大，则抽检比例大。 ◆ 厂商信誉——信誉好，则抽检比例小。 ◆ 生产技术——水平高，则抽检比例小。 ◆ 储存时间——时间长，则抽检比例大。
● 实物验收	<ul style="list-style-type: none"> ◆ 外观质量验收——检验外包装是否完好。 ◆ 数量验收——点件法、抽验法。 ◆ 重量验收——检斤验收法、约定重量法。 ◆ 精度验收——金属材料尺寸精度检验。

（三）搬运作业的标准化

验收完毕，产品便可搬运入库内进行储存，这是各环节之间能否形成有机联系和紧密衔接的关键。大部分人默认搬运的存在，却不设法消除它。有些人想到用输送带的方式来克服，这种方式仅能称之为花大钱减少体力的消耗，但搬运本身的浪费并没有消除，反而被隐藏了起来。在此我们利用搬运流程标准化来消除搬运浪费。

搬运的标准流程如图 3-8 所示：

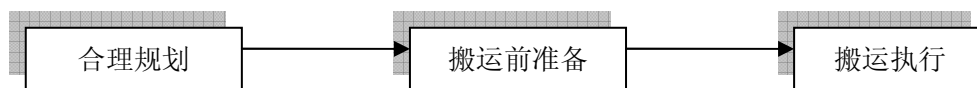


图 3-8 搬运流程图

1. 合理规划

在组织搬运前，首先要有良好的规划，才能避免出现混乱、浪费等现象，还能事先解决当中的许多问题。从案例我们得出：安得的仓库面积较大，单次行走距离约为 80 米，移动经济较低。由于仓库内使用垫仓板，不能用于搬运，因此未能实现单元化作业，每次移动的数量也受到限制。但是，种种问题，我们可以通过合理规划组织搬运提高搬运效率，如下：

从搬运距离、搬运空间、搬运时间 3 个因素进行探讨考虑，如表 3-5 所示：

表 3-5 搬运因素考虑说明表

考虑因素	注意要点
(1)搬运的距离	设法运用最低成本、最有效的方法来克服搬运位移、长度，以尽速将对象送到指定的场所。
(2)搬运的空间	企业在规划时也许已经预留搬运空间，如通道、进出口等。但是，空间的要求受搬运系统的效率影响很大，为了防止堵塞，则需要提高系统运行效率。
(3)搬运的时间	避免“过快”（会影响后续作业效率），或不及（往往会增加仓储成本）的情形发生。必须配合适当的机械和运作方式，才能使得对象在恰好的时间到达确实的位置。

（1）搬运的距离

① 存在问题：单次搬运距离长。

② 解决办法：

a) 合理选择保管场所；

b) 电视机、空调等进出库量大、频率多的，尽量放在靠近出货区；

c) 进出库量小、频率低的，放在离出货区较远的地方。

（2）搬运的时间

① 存在问题：搬运速度过慢作业时间长，尤其是在旺季，常常引起顾客投诉。但有不能过快，尤其是设立拣货区后，否则，一方面容易造成货损，另一方面会造成拣货区货物堆积。

② 解决办法：配合适当的机械和运作方式，尽量使总体操作有条不紊。

（3）搬运的作业环境

① 存在问题：作业环境管理不善——倾倒、淋湿的货物，栈板使得现场混乱，而影响装卸搬运的速度，还容易造成事故。

② 解决办法:

- a) 安排专门的后勤人员, 负责清理现场;
- b) 仓管员巡视工作现场时指出混乱的地方, 区分出要与不要的物品;
- c) 不要的移出工作区域, 必要的规范放置并注明标志;
- d) 区分通道和非通道的区域, 两边用粗黄线, 作业中通道不得滞留物品, 特殊情况需要在通道上放置, 要摆放“通道临时启用”标牌、消防通道等禁止存放货物的, 在地上标上符号。

建立良好的工作环境, 使得让工作顺畅, 让员工拥有良好的工作心态, 进而提高工作效率。

2. 搬运前准备

(1) 合理安排货车与拣货区之间的作业人员的数量。

根据装卸搬运的不同物品类别, 制订出装卸搬运作业人员的定额标准, 以避免作业人员出现过度疲劳的情况。

(2) 选择合适的搬运设备。

根据要搬运的货品类型及数量, 来选择合适的搬运设备, 同时在搬运过程中要尽量防止货物移动过程中发生倾倒、碰撞等现象, 降低货损。

① 小车的价格较低, 每次装载量较大。由于安得公司单次出入库搬运的距离较长, 因此提高单次装载量能够提高搬运效率。所以, 安得公司在 30 米以上的中长距离的搬运中应该尽量使用小车。

② 电视机等大件产品, 可以选择库内的柴油叉车进行作业。进行保管位置的选择时, 电视机应该安排在靠近出货区的储位, 移动距离较短, 因此使用叉车作业是可行的。

(3) 解决与设备相关的问题。

① 使用叉车作业时, 为了避免叉车叉过头或叉不到位, 装卸之前, 先测量出货物的长度, 然后计算出叉车的叉套应该插入的位置, 标上明显的标志作为警戒线, 叉取货物时应该刚好叉到所标明的位置。

② 搬运过程中, 货物容易碰撞倾倒。在小车四周加上灵活可拆卸的防护栏, 降低倾倒的危险, 小车上的堆垛也就可以达到 3 层。防护栏还可以提高搬运过程的运行速度。

③ 准备好有关的工具。在货车到达之前, 根据实际情况, 判断是否需要垫仓板、遮雨布。如有需要, 则垫仓板和遮雨布集中要放在出入口处, 以免需要的时候去寻找而浪费时间。同时, 遮雨布要配有夹子, 防止被风吹起, 淋湿货物。

3. 执行搬运的要点与说明

(1) 注意货物是否禁止倒置、需要小心轻放。

(2) 货物装上小车前把小车的方向摆好, 提高搬运的效率, 同时也要避免转弯时货物由于离心力而倾倒。

(3) 被撞倒、淋湿的货物需要隔离, 应该由后勤人员在仓管员的指导下进行, 搬运工人继续搬运, 不要中断工作。

(4) 包装已经变型的, 先放置一边, 由仓管人员根据情况进行处理。

3.3.2 在库管理的标准化

产品入库后, 将有长期或短期的在库保管。在库管理的目标包括: 空间的最大化利用、劳力及设备的有效使用和货品的妥善保管。为了达到这些目标, 每一

环节都要做好计划安排。

在库管理的主要流程如图 3-9 所示：

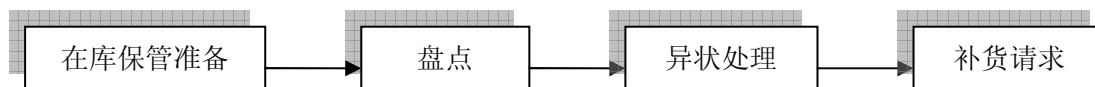


图 3-9 仓储在库管理流程图

接下来，我们对在库管理流程中的盘点、异状处理、补货请求的具体运作设立标准化。

（一）盘点流程的标准化

盘点是为了确定现存量，并修正错误信息。为了核查货物，使出入库的管理方法和保管状态变得清晰，通过盘点发现问题，从而提出改善的方法。

盘点的流程如图 3-10 所示：

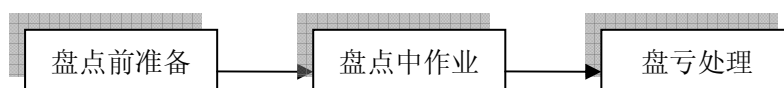


图 3-10 盘点流程图

具体操作流程说明：

1. 盘点前准备

为了在较短时间内利用有限的人力完成盘点的工作，首先要做好充分的事前准备，包括：决策准备（盘点的程序方法、准备好相关资料）、人员准备、盘点工具准备、盘点前指导。

决策准备：确定盘点频率如图 3-11 所示：

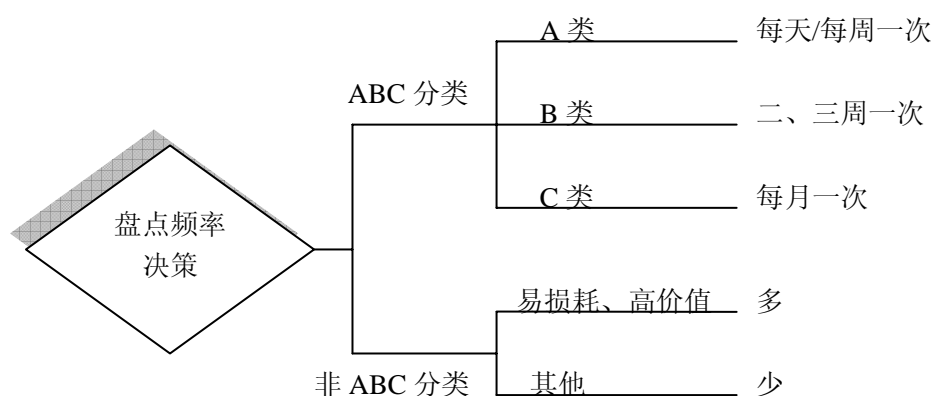


图 3-11 盘点频率决策

为了防止过久的盘点对公司造成损失，但又考虑到可用资源的限制，对于采取 ABC 管理的企业，建议 A 类货品每天或每周盘点一次，B 类货品每两三周盘点一次，C 类货品每月盘点一次。如果没有实行 ABC 管理的，对容易损耗毁坏及高单价货品适当增加其盘点次数。

安得公司的一个合作 5 年之久的客户——T 客户是一个生产销售电饭煲、电

磁炉、电炖锅、电水壶等产品的制造企业，并且每年的产品有 30% 以上的增长。由于大部分小家电产品不属于高价值的 A 类，但 T 客户又占了安得公司很大份额的业务，所以应当将其大部分产品划入 B 类进行管理，而将其价值较高的高智能化的产品划入 A 类。

2. 盘点中作业

依照先仓库，后冷藏库，最后卖场的顺序进行盘点。

3. 盘亏处理

盘点结束以后，发现所得数据与信息系统的相符时，追查差异原因。原因查明后，针对主要原因适当调整处理。

另外，盘点选择在财务结算前期或淡季进行。为了确保盘点正确性，除了人员培训加强教育外，工作进行期间应该加强指导与监督。

(二) 异状处理作业的标准化

1. 根据异状原因采取不同的处理办法，如表 3-6 所示：

表 3-6 异状原因处理表

异状原因	处理办法
储存条件和保管措施出问题	主动改善储存条件。
商品生产工厂方面问题	联系委托单位，协商处理，积极配合解决。

2. 根据异状程度采取不同的处理办法，如下表 3-7 所示：

表 3-7 异状程度处理表

异状程度	处理办法
轻微异状	用“商品异状问题报告表”填他委托单位掌握，仓库加强保管、检查措施，注意其变化发展情况。
重大异状	填报“商品异状问题催请处理表”送委托单位催请处理。

(三) 补货请求流程的标准化

1. 补货请求流程，如表 3-8 所示：

表 3-8 补货请求的流程

流程	具体操作要点
● 取得数据	取得盘点后的数据及异状处理后货物情况的相关数据。
● 选择模型	根据货品的类型选择相应的公式或模型。
● 确定补货量	用公式或模型计算并确定应该补货的货品和数量。
● 发出请求	填写补货申请表，发出补货请求。

补货请求的流程虽短，但每一步都很重要，尤其是选择模型和确定补货量方面，在下面 3.4.2 中问题的解决部分，可以利用这个标准化流程来解决主动持续补货的问题。

2. 补货请求流程标准化的注意事项

(1) 确定补货方式

补货作业的发生与拣货区的存货量紧密相关，采取何种补货方式，要视拣货区存量的情况而定，以避免拣货中途才发现拣货区货量不足，而影响整个出货作业。补货方式有以下三种，公司应根据具体情况而定，如表 3-9 所示：

表 3-9 补货方式表

拣货情况	补货方式
一日内作业量变化不大的, 紧急插单不多, 或是每次拣取量大需事先掌握的	批次补货
拣货时间固定, 且公司出力紧急事件亦固定的	定时补货
每次拣取量不大, 紧急插单以及一日内作业量不易掌握的	随机补货

(2) 充分利用信息系统

当库存管理业务中使用信息系统时, 有关库存品的记录都列入出入库的记录整理、余额的记录、订购点的确认、订购日期、订购数量、入库及出库预定日、请求出库传票的发行及日程管理等等, 这些信息处理的繁琐操作均可交由计算机处理。

进行计算机库存管理时, 有以下四个要点:

- ① 将所处理的物品符号化(用数字或字母符号来代替物品名称的整理号码);
- ② 编定区划(如排数、列数、层数等);
- ③ 绘制作业流程图, 并进行试验、实施。

3.3.3 出库管理的标准化

收到出库单后, 相关人员便要准备好出库作业。相关人员要严格按照出库业务的一般程序进行出库, 遵守出库的各项规定, 正确处理出库业务中的各种问题, 从而提高出库业务的效率和准确率。

出库标准化后的流程如图 3-12 所示:

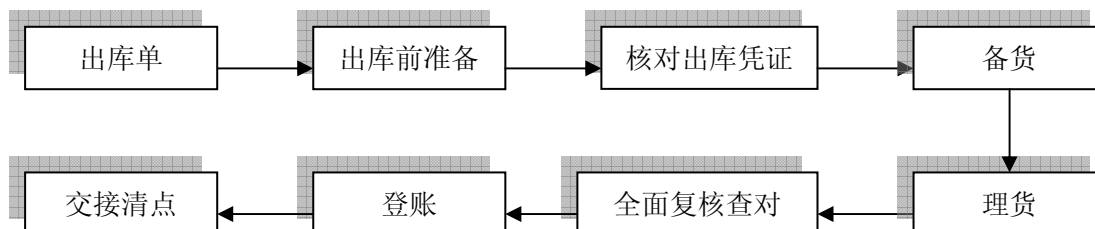


图 3-12 仓储出库管理流程图

接下来, 我们对出库管理流程中的出库前准备、备货、理货的具体运作设立标准化。

(一) 出库前准备作业的标准化

完成以下出库前的准备工作:

1. 对货物原件的包装整理。
2. 零星货物的组配、分装, 尤其是安得公司的零担运输部分。
3. 包装材料、工具、用品的准备。对从事装、拼箱或改装业务的仓库, 在发货前应根据性质和运输部门的要求, 准备各种包装材料及相应的衬垫物, 以及刷写包装标志的用具、标签、颜色和钉箱、打包等工具。如电视机等不能倒转放置的产品, 要用明显的标志表明。
4. 待运货物的仓容及装卸机具的安排调配。商品出库时, 应留出必要的理货场地, 并准备必要的装卸搬运设备, 以便运输人员提货发运或装箱送货, 及时装载货物, 提高装车效率。

5. 发运作业的合理组织。发运作业是一项涉及人员较多，处理时间较紧，工作量较大的工作，进行合理的人员组织和机械设备的协调安排是完成发货的必要保证。

由于出库工作比较细致复杂，工作量也大，事先对出库作业加以组织，安排好作业人员和机械设备，保证各环节紧密衔接。

(二) 备货流程的标准化

备货按照以下的标准流程进行，如表 3-10 所示：

表 3-10 备货流程

流程	执行要点
● 销卡	先销卡、后付货
● 理单	根据出库单的货位，按出库顺序排列
● 核对	“以单对卡，以卡对货”，进行单、卡、货三核对
● 点数	点清应付的数量，防止差错
● 批注地区代号	为方便下道作业环节，外包装上必须批注地区代号
● 签单	保管员逐笔在出库签名和批注结存数，以明责任且供登账时进行账目实数的核对

(三) 理货流程的标准化

理货按照以下的标准流程进行，如表 3-11 所示：

表 3-11 理货流程

流程	执行要点
● 核对	逐车逐批地进行，确保单货数量、品唛、去向完全相符
● 置唛	在外包装两头，字迹清楚，不错不漏
● 待运装车	按地区到站逐批装车，防止错装、漏装

以上我们着重阐述了入库管理中的保管场所安排、入库前准备、装卸搬运；在库管理中的盘点、异状处理、补货请求；出库作业中的出库前准备、备货、理货这些部分。下面，我们将运用所设计的仓储运作流程标准化方案解决安得公司设立拣货区、主动持续补货、绩效考核这三个突出的问题。

3.4 安得公司仓储问题解决方案

3.4.1 对“仓储设计”问题提出的解决方案

(一) 拣货问题的分析

客户、承运司机常抱怨等待时间太长，仓储作业效率太低，主要表现为装卸车的间隔时间长，装卸工人与工具需要在不同的仓库之间调来调去，车辆排队现象严重。产生这种现象的原因可以归结为作业流程安排不合理，原来采取的流水作业，导致了交叉等待。单次出入库行走距离长，车辆对位、仓管员找库位、人员器具组织又花费许多时间。如果打破流水作业，在车辆到达之前在出货区准备好货物，车辆到达马上装车，则可以减少车辆等待的时间。

（二）用标准化流程提出解决方案

1. 方案简介

在案例中，有仓库管理专家就提出了通过设立拣货区来解决其中的一些问题，这是比较可行的，另外我们还可以通过制订一系列的标准化运作流程来使其实行过程更优、更规范化，有章可循。根据案例中的情况，我们可以设立两个拣货区，即在 3 仓库和 4 仓库分别划出一个区域作为拣货区。在实际操作中，根据具体情况（种类、数量、体积、重量）来安排两仓库间的拣货与配装。

2. 拣货区具体运作流程如图 3-13 所示：

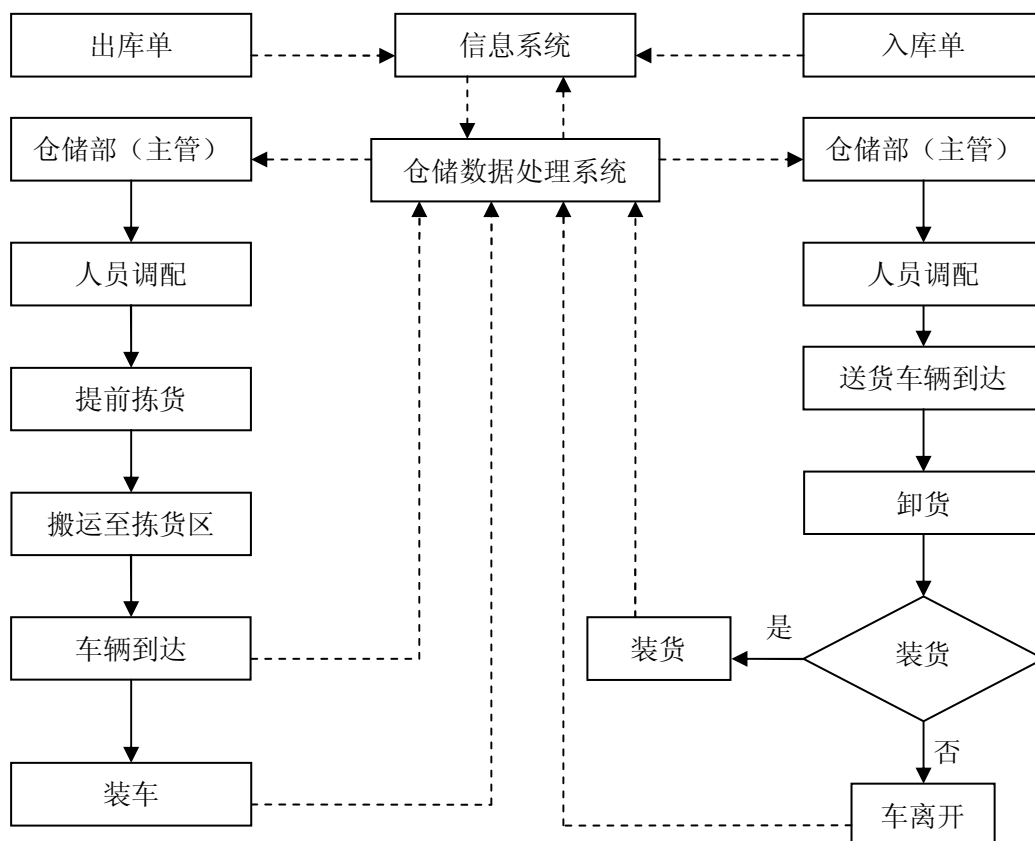


图 3-13 设立拣货区后的作业流程图

3. 标准化操作流程说明

（1）装车作业流程：

①信息系统和仓储数据处理系统收到出库单，并进行处理，决定调用哪些车辆，根据货品种类、数量、体积和重量要将货品临时放在哪个拣货区。例如，需要装车的货物的类型差不多，且 70% 以上的货物是存放在 3 仓库的，则应将预定的货物全部放在 3 仓库的拣货区，另外 30% 的货物在 4 仓库拣取后也要搬运到 3 仓库的拣货区中。

②信息系统将订单处理结果交给仓储部，由仓管人员具体安排搬运工的调配，根据车辆预定到达时间提前派人进行作业。

③当运输车辆到达时，仓储数据处理系统立即通知仓储部，仓储部指挥车辆停在指定的拣货区，派事先调配好的搬运工到进行装车。

装车作业流程标准化效果：原先需要 6-7 个装卸工花 70 分钟的装车工作，就只需 3 个装卸工用 40 分钟就能完成，能够有效地减少了车辆的等待时间，而且车辆只要排一次队就可以了。

(2) 卸车作业流程:

①信息系统和仓储数据处理系统收到入库单, 并进行处理, 根据货品种类、数量、体积和重量以及应该存放的仓库位置, 决定临时放在哪个拣货区。

②订单处理员将订单处理结果交给仓储主管, 当运输车辆到达时, 仓储数据处理系统立即通知仓储部, 仓储主管派调度员指挥车辆停在指定的拣货区。

③仓储主管派事先调配好的搬运工到进行卸车, 如果货品还需要装到另外的车辆则由调度员指挥相应的车辆过来, 如果货品需要存放则仓储主管调配人员将货品搬运入仓库。

卸车作业流程标准化效果: 原先需要 6-7 个装卸工花 60 分钟的装车工作, 就只需 3 个装卸工用 25 分钟就能完成, 能够有效地减少了车辆的等待时间。

通过设立拣货区并进行标准化的流程操作, 就可以打破流水作业, 减少了车辆交替等待的可能性。

3.4.2 对“主动持续补货”问题提出的解决方案

(一) 补货问题的分析

在 C 客户、安得公司、V 公司三者中, 暂时只能实现 C 客户通过安得公司由生产基地向各销售公司进行主动的持续补货, 不能实现 C 客户通过安得物流向 V 公司实现持续补货。V 公司能根据其卖场上报的销售信息以及总公司记录的卖场现有库存量, 对各单位的主动补货, 但这种主动持续补货模式只是在公司内部实行。这主要是因为 V 公司与 C 客户或安得公司之间还没有实现信息系统的对接, 不能做到数据同步更新而导致的。因此, 需要采取一种简单易行的方法, 实现 V 公司与 C 客户、安得公司的信息系统对接。

(二) 用标准化流程提出解决方案

1. 方案简介

根据安得公司的实际情况, 我们可以采用类似于 (ATP, Available to promise inventory) 的主动持续补货模型, 将主动持续补货的流程规范化。

运用可承诺库存的运营模式, 明确了仓库中可以用来答应顾客在确切时间内供货的产品数量。主要针对新订单, 通过对主动持续补货这个模型的标准化操作, 只要供需双方均采用这个模型进行预测与操作, 就可以取得一致的数据, 如需求预测、顾客订货量等, 不需要实现公司间信息系统的对接。从而打破了安得公司无法与客户实现系统对接时的信息不对称的状况, 能够促进安得公司与其他客户的合作。

可承诺存货: 指还未被预订的供货能力, 可以承诺提供给新的客户需求。

补货请求流程如图 3-14 所示:

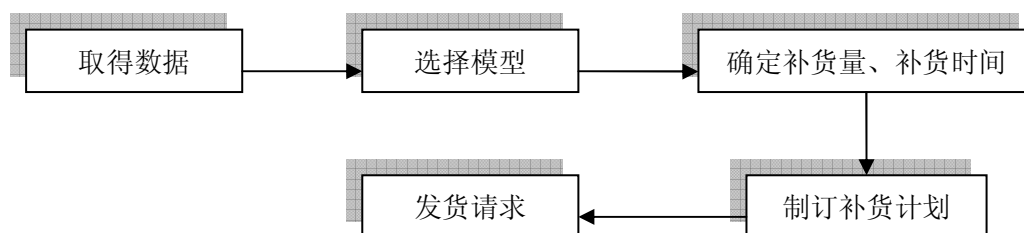


图 3-14 补货请求流程图

2. 补货请求的流程说明，如表 3-12 所示：

表 3-12 补货请求的流程

流程	具体操作要点
● 取得数据	取得盘点后的数据及异状处理后货物情况的相关数据。
● 选择模型	根据货品的类型选择相应的公式或模型。
● 确定补货量	用公式或模型计算并确定应该补货的货品和数量。
● 发出请求	填写补货申请表，发出补货请求。

3. 进行主动持续补货的具体操作如下：

(1) 取得数据：期初库存、需求预测、顾客订货量、主动补货量。

(2) 根据货品的类型选择相应的公式或模型。

制订主动补货量的基本模型（步骤）：

①计算计划持有库存（POH）

$$I_t = I_{t-1} + P_t - \max(F_t, Co_t)$$

I_t ：t 期末现在库存量； P_t ：t 期主动补货量；

F_t ：t 期预计需求； Co_t ：t 期顾客订货量。

②决定订货量和订货时间

订货量：计划期总订货量和每次批量。

订货时间确定原则：主动补货量和补货提前期应保证现在库存量非负。

(3) 计算可承诺库存（ATP）得出结果，确定主动补货的时机。

第一周：

ATP=期初库存+本周主动补货量 - 直到下一次主动补货量到达为止的全部订货量
以后各周（只有到了有主动补货量时才计算）：

ATP=该周主动补货量-下一次主动补货量到达为止的全部订货量

例如，假设期初库存为 64，每次主动补货量为 70，其需求预测和顾客订货量如下图所示，则相应的主动持续补货操作如表 3-13 所示：

表 3-13 主动持续补货预测表

	六月				七月			
	1	2	3	4	5	6	7	8
需求预测	30	30	30	30	40	40	40	40
顾客订货量	33	20	10	4	2	0	0	0
现有库存量	31	1	41	11	41	1	31	61
主动补货量			70		70		70	70
可承诺存货	11		56		68		70	70

则 1-2 周内还可满足 11 个单位的额外的订货量，3-4 周可满足 56 个单位，5-6 周可满足 68 单位，7、8 周都可以满足 70 个单位的额外订货量。

(4) 制订出补货的计划并准确执行。

运用可承诺库存的运营模式，能明确仓库中可以用来答应顾客在确切时间内供货的产品数量。主要针对新订单，通过对主动持续补货这个模型的标准化操作，只要供需双方均采取这个模型进行预测与操作，就可以取得一致的数据，不需要实现公司间信息系统的对接。从而打破了安得公司无法与客户实现系统对接时的信息不对称的状况，能够促进安得公司与其他客户的合作。

3.4.3 对“花城分公司”问题提出的解决方案

（一）花城分公司的问题分析

在案例中提到，由于存在跨库管理，花城分公司有大批量的老员工离职，新招人员 60% 工作不到一周就离职，装卸人员不稳定，数量不足，移库后整个工作被打乱。员工的士气变得低下，所以纷纷离职。在这种情况下，必须做好人员的管理，除了有组织有秩序的工作安排，更需要一套公平合理的绩效评估和激励机制来留住员工，提高士气。另外，仓储管理员的工作也要加强，避免混乱，提高效率。

（二）用标准化流程提出解决方案

1. 方案简介

公平合理的绩效评估和激励机制能更好地留住员工，但是，由于各种主观或客观的原因，要做到公平合理并不容易，因此，绩效评估制也需要制订一个标准化的运作流程并严格地按照这个流程执行，保证公平公开，才能让众人所了解、信服和支持，从而起到留住员工激励员工的作用。同时，仓管员是企业仓储作业中很重要的角色，因此，有必要对仓管员的每日工作流程实行标准化，确定每天的工作内容，这样就能避免由于仓管员疏忽而造成的混乱。

2. 绩效评估具体运作流程，如图 3-15 所示：

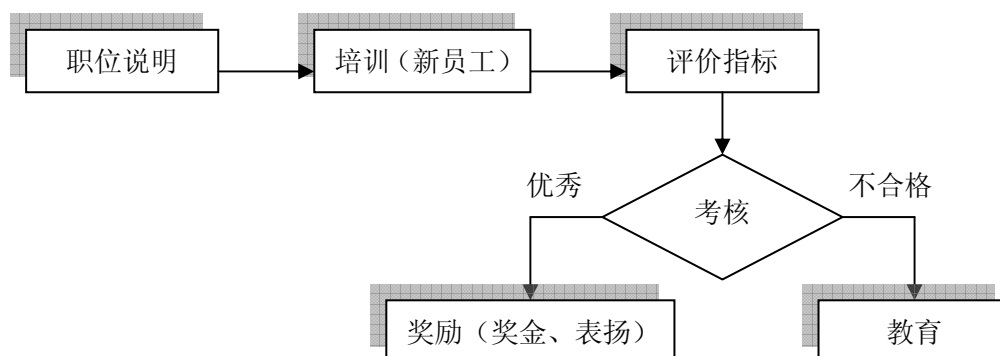


图 3-15 绩效评估流程图

首先，招员工之前，要有清楚的职位说明，以免员工招进来以后因为不喜欢而离职。接着，新员工进来之后要进行培训，熟悉工作以后，就可以走上工作岗位，然后按照评价指标对其进行评估。

根据安得的实际情况，我们制订一些评价指标，考核对象：作业人员、后勤人员，具体的工作表现评估如表 3-14 所示：

表 3-14 工作表现评估表

项目 \ 分数	1	2	3	4	5
工作品质（工作成果如何，有无造成货损或虚报的情况）	常有错误	成果低于一般标准	尚可满意	成果常在标准以上	极佳，常令人满意

可靠性(是否坚守本职工作, 可信赖)	无规则, 不能信赖	时常疏忽	尚可信赖	确实可靠	相当可靠
工作知识(对工作必需之学识程度, 工作知识如何)	常识不够	学识有限	中等	颇佳	常识丰富, 并能深明要点
工作完成质量	完全达不到标准	勉强达到标准	基本达到标准	完成质量比较好	一直高质量完成
对突发事件的应对能力	处理很差	处理比较迟缓	中等	较快应对	很快高效处理
主动性(主动负责指定工作内的任务, 或监督他人共同做好)	完全被动	很少表现主动	偶有表现	比较主动	非常主动地负责
合作性	素不与他人合作	与他人合作有困难	与人共同工作正常	有团队精神	非常热心尽力与人合作
考勤	时常缺席	常借故缺席	较少缺席	很少缺席	全勤
工作效率	习惯拖延效率极低	经常拖延效率较低	偶尔拖延效率中等	很少拖延效率较高	很不拖延效率很高
操作标准化程度	很低, 基本不按要求做	较低, 离要求还有一些差距	中等, 基本达到要求	较高, 较好地达到要求	很高, 完成达到要求

由仓管人员根据上表对搬运装卸工进行评分考核, 每周一进行一次, 并对前十名进行公布表扬, 并给连续四周上榜的工人颁发奖金, 以激励所有工人积极地做好工作。

3. 仓管人员每日作业流程的标准化, 如表 3-15 所示:

表 3-15 仓管人员的每日工作安排表

工作阶段		工作事项
工作开始前	准备阶段	<ul style="list-style-type: none"> ● 开启现场大门, 更换工作服、鞋帽、佩戴工作牌 ● 清洁四周环境 ● 开通办公设备
	开早会, 安排事项	<ul style="list-style-type: none"> ● 简要向大家说明昨天的工作状况, 通报问题对策的进展情况。 ● 说明当天的工作安排, 包括人员、工具安排

工作 进 行 中	确定工作开始	<ul style="list-style-type: none"> ● 确认工作人员是否已经进入岗位 ● 设备是否按要求点检 ● 所需工具是否已经准备好
	随机确认工作质量	<ul style="list-style-type: none"> ● 是否按照标准进行作业 ● 检查换岗人员的作业是否正确 ● 对不熟练者进行指导 ● 开展轮训，尽量使每个人都成为多面手 ● 仔细观察，改善作业，杜绝白干、瞎干和野蛮作业 ● 欲改善的部分超出管辖范围，寻求上司援助 ● 统计改善实绩，报告上级
	随机确认库存情况	<ul style="list-style-type: none"> ● 检查存货，及时发现问题 ● 按照先进先出的原则 ● 盘点库存资料及时录入
	随机处理货损	<ul style="list-style-type: none"> ● 按照规定的途径、时间处理完毕 ● 做成《货损报告》上报，对做出对应决策 ● 必要时，寻求外部援助来解决
	确保作业进度	<ul style="list-style-type: none"> ● 车辆人员未能及时到位，适当调整 ● 未能及时完成任务时，紧急对策 ● 工作量高于预定目标，相应准备没做好，立即挽救 ● 准时完成装卸搬运作业 ● 工作实绩可见化，情报共享
	随机推行新方法	<ul style="list-style-type: none"> ● 率先遵守《5S 规定》，检查部下执行情况 ● 结合实际，加大对标准的指导力度
工作 结 束	确认任务完成情况	<ul style="list-style-type: none"> ● 额定工作是否已经按时完成 ● 安排加班事项 ● 作业台上有无滞留物品 ● 填写仓储报告 ● 清洁工作环境，做好五防（防尘、防火、防水、防虫、防盗） ● 做好第二天的工作准备 ● 关闭所有动力电源 ● 锁上大门，放好钥匙，离开
	其他	<ul style="list-style-type: none"> ● 参加各种会议，充分发表意见，并记下要点 ● 解决突发事件，必要时紧急联络上司，寻求援助 ● 参观学习其他仓库的管理的长处，善加引用 ● 与下属面谈，交换意见 ● 拟定各种新计划

绩效考核合理有效能够提高员工的忠诚度，提升员工的工作积极性，从而提高企业的效益；仓管员的规范化操作能够提高企业的仓储管理水平，运作流程标准化为两者的实施都提供了帮助。

4 配送流程标准化设计

4.1 安得公司配送现状的分析

配送是安得物流公司提供服务的过程中直接面向末端客户的环节，配送的响应速度、配送货物的准确性都会直接影响客户服务水平。“精益配送”是安得公司近几年努力发展的方向，在信息系统、作业方法上，安得已经做出很大的改进。可是在一些作业流程上，安得公司还是存在一些问题：

1. 下订单时间不统一，订单大多为多批次少批量类型，很难实现共同配送；
2. 每个客户配送区域不集中，很多配送路线无法进行集拼；
3. 满载率低，零担运输比率高；
4. 信息反馈不及时，逾期率高，配送回单不完整，批量回单遗失等现象。

根据分析，我们可以把上述问题用鱼骨图来分析，如图 4-1 所示：

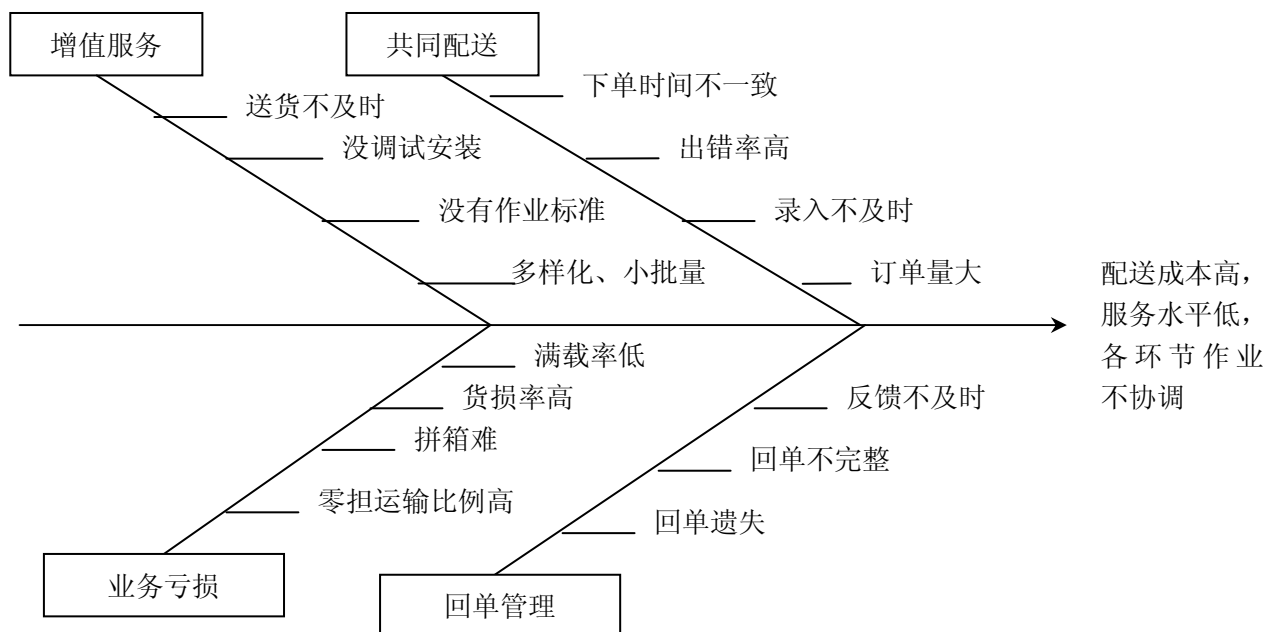


图 4-1 配送问题鱼骨分析图

根据上面的鱼骨分析图，可以看出安得公司在现阶段的配送流程中存在的问题主要是集中在订单处理、配装作业、回单管理、增值服务方面。由于各个作业环节的衔接不协调，造成作业效率较低，服务水平较低，成本高的问题。

4.2 配送流程标准化设计概述

4.2.1 配送流程标准化设计的目标

配送流程标准化分别从订单处理、备货、分拣与配货、配装、运送和结算等方面流程的标准化来实现快速响应的目标，以达到高效、准确地处理订单，合理包装和保护货物，提高车辆装载率，降低配送单位成本。提高安得公司的核心竞争力，为顾客提供更优、更全面的服

4.2.2 配送流程标准化设计的基本内容

我们结合案例中配送流程中存在的问题制订标准化的运作流程，包括订单处理、备货、分拣及配货、配装、送货及结算流程的标准化。并且对具体的流程进行说明，其中还包括在运作流程中处理问题的标准和原则。

我们根据安得公司配送过程中存在的问题如“共同配送”、“增值服务”、“淡旺季作业”等，从流程的角度提出了标准化的运作流程的解决方案。

4.3 配送流程标准化的具体设计

配送管理的流程如图 4-2 所示：

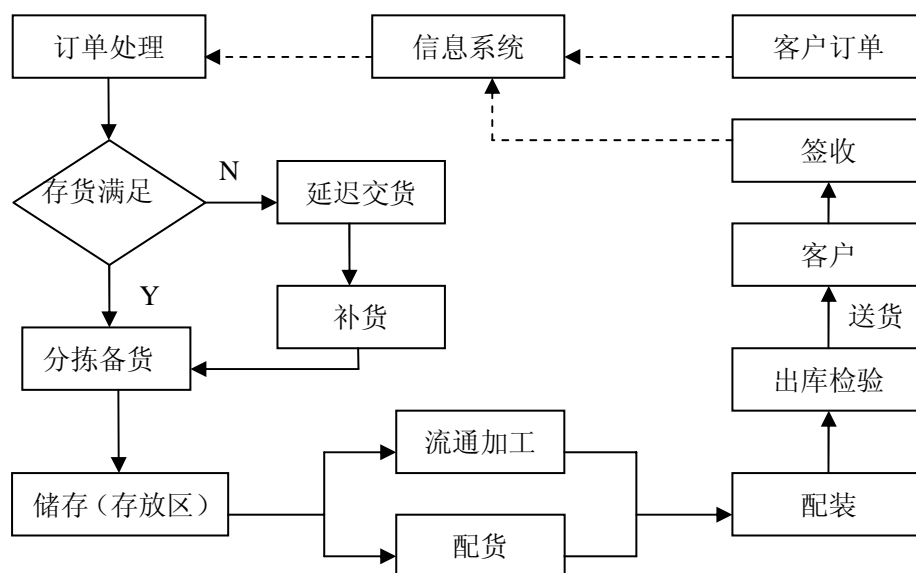


图 4-2 配送管理流程图

下面我们将从订单处理、备货、分拣和配货、配装、运送和结算等方面说明配送标准化的运作流程。

4.3.1 订单处理流程的标准化

（一）订单处理的标准流程

订单处理是实现顾客服务目标最重要的影响因素。改善订单处理流程，缩短订单处理周期，提高订单满足率和供货的准确率，提供订单处理全程跟踪信息，可以在提高顾客服务水平的同时降低物流总成本。如图4-3所示：

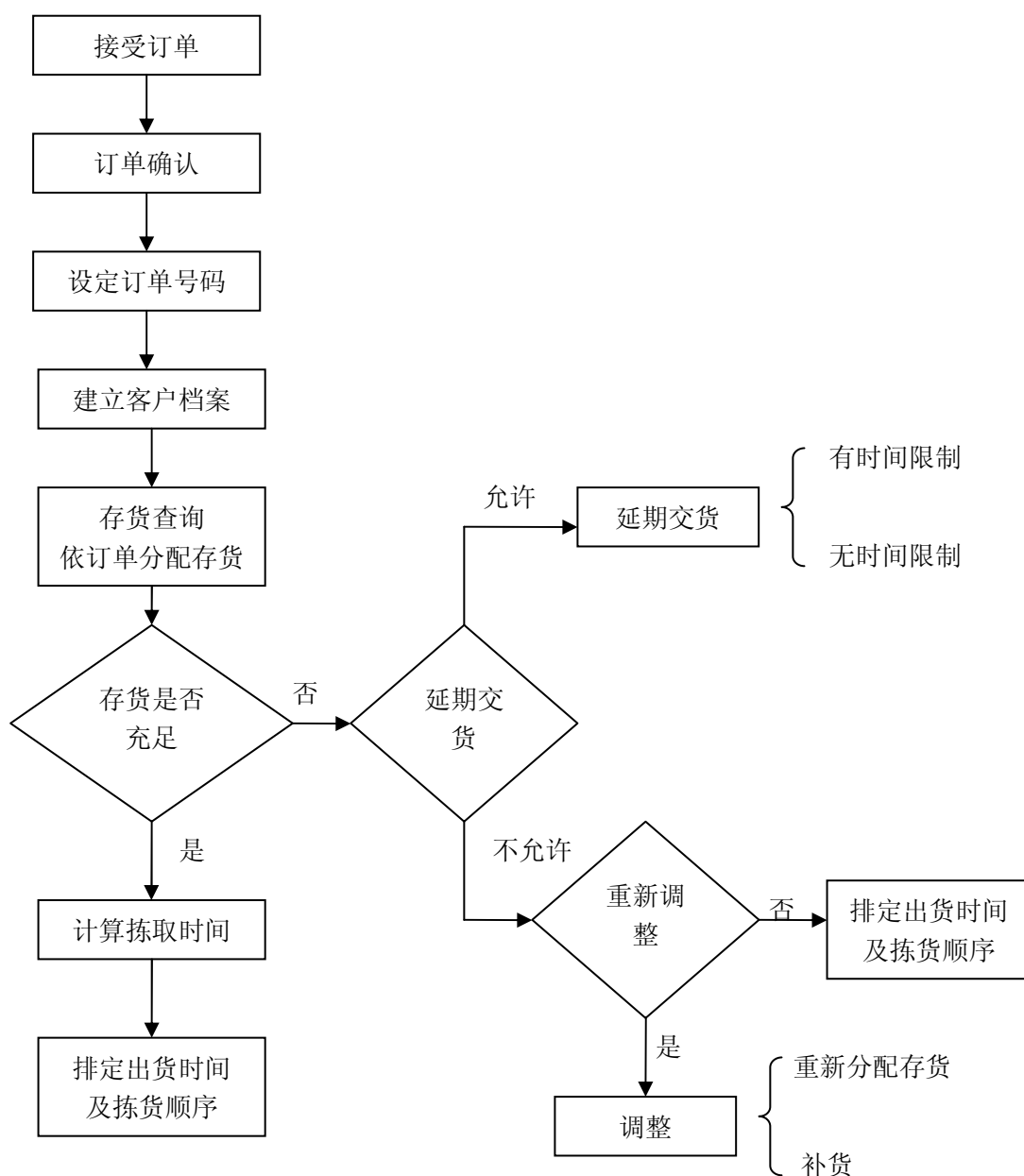


图 4-3 订单处理流程图

安得公司的信息系统是运作的核心，是公司与供应商和客户进行有效沟通的媒介，公司在接受订单时通常是通过电子数据传递、电话、传真等方式接受用户的订货资料。

订单的接受与录入效率直接影响到作业的准确性与及时性，所以，在订单处理时，要统一订单格式的标准和业务人员的作业，例如：订单必须与订单处理软件的界面保持一致，这样不但便于操作，还能达到提高订单录入效率，降低出错率的效果；订单录入员应该及时地把收到的订单信息录入避免订单处理滞后。

（二）订单处理员作业流程的标准

安得公司在订单处理的过程中出现订单录入不及时、易出错的问题。从订单处理员的作业流程可以看出，很多订单处理出现的问题都跟订单处理员有关，所以制订一个订单处理员的标准化操作对提高作业的效率 and 准确率十分重要。

根据订单处理员的业务我们提出订单处理员的作业流程，如表4-1所示：

表 4-1 订单处理员作业流程

流程	作业
流程一	接受订单
流程二	确认订单
流程三	订单分类
流程四	设计订单档案资料内容
流程五	输入订单资料
流程六	存货查询及订单分配存货
流程七	订单数据处理输出

流程一 接收订单

通过电话、传真或电子数据传递等方式接受用户的订货资料。

流程二 确认订单

接收到顾客订单以后，首先对用户的信用进行确认，看其应收帐款是否已经超过信用额度，以确定继续或停止输入该订单。当订单通过信用检查后，还要继续确认订单的其他基本内容，包括订货的种类、数量、配送时间、价格、包装等。如图4-4所示：

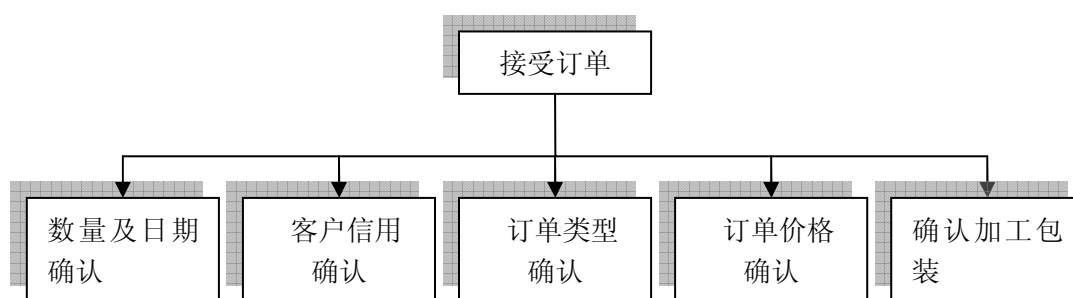


图 4-4 订单确认

接受订单后，需对其内容进行确认，主要包括以下几点：

（1）确认货物数量及日期。货物数量及日期的确认是对订货资料项目的基本检查，即检查品名、数量、送货日期等是否遗漏、笔误或不符合公司要求的情形。尤其当送货时间有问题或出货时间已延迟时，更需要与客户再次确认订单内

容或更正运送时间。

(2) 确认客户信用。不论是何种订单，接受订单后都要查核客户的财务状况，以确定其是否有能力支付该订单的账款。通常的做法是检查客户的应收账款是否已超过其信用额度。

(3) 确认订单形态。对不同的订单形态采取不同的交易及处理方式。

(4) 确认订货价格。不同的客户、不同的定购量，可能有不同的价格。输入价格时订单处理员应运用系统对其加以核查。若输入的价格不符（输入错误或因业务员降价接单等），则加以锁定，以便主管审核。

(5) 确认加工包装。确认客户对订购的商品是否有特殊的包装、分装或贴标签等要求，或是有关赠品的包装等资料都应确认记录，并在订单上注明客户的出货要求。

流程三 订单分类

确认订单后，要根据订单类型对订单进行分类处理以提高作业效率，订单分类原则分两种情况：

(1) 特殊订单，如：缺货补货订单、延迟交货订单、紧急订单或远期订单，应确定这些订单的优先处理。然后根据客户的信用度、货物批量和交货期限做优先处理。

(2) 一般订单（要求3~5天送达），则根据客户的信用度、货物批量和交货期限做优先处理。根据客户的货物批量大小来把客户分成大、中、小客户，其中大客户的货物多，数量大，可以取得规模效应，配送的单位成本低。在实际操作中，应该先处理大客户再处理小客户，由不同的小组对这些单据分类审核，确保高质量的客服水平。如图4-5所示：

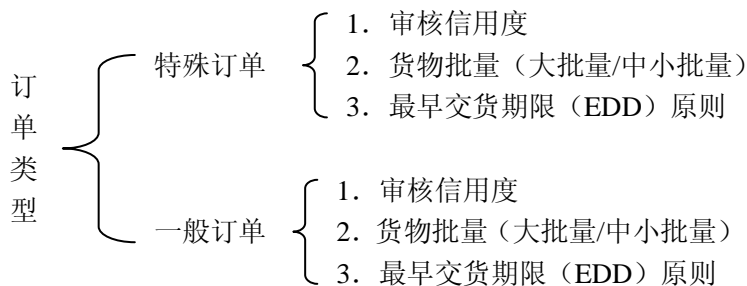


图 4-5 订单处理原则

流程四 设计订单档案资料内容

订单分类后，建立一个完整的用户订单档案，以便于本次交易的进行和以后与用户的长期合作。根据实际需求设计订单档案资料内容，以符合后续作业所需。另外，用相关字段关键词把订单表头文件与订单明细文件加以连接，其中表头文件用于记录订单的整体性资料，如订单单号、订单日期、客户代号等，订单明细文件则记录每笔订货品品种详细资料，如商品代号、商品名称、单价等。如表4-2所示：

表 4-2 客户档案

客户资料项目	说明
客户名称、代号、等级等	
客户信用额度	对批发用户或第三方物流的客户。
客户销售付款及折扣率的条件	对批发用户或第三方物流的客户。
开发或负责此客户的业务员资料	经营批发业务的配送中心或第三方物流的客户。
客户配送区域	
客户收帐地址	
客户点配送路线顺序	根据区域、街道、客户位置，将客户分配于适当的配送路径。
客户点适合的送货车辆状态	客户送货上门点的街道有车辆大小的限制，因此须将适合该客户的车辆类型记录在系统中。
客户点卸货特性	客户所在地点或客户位置，由于建筑物本身或周围环境特性（如地下室有限高或高楼层），可能造成卸货时有不同的卸货需求，在车辆及工具的调度上须加以考虑。
客户配送要求	客户对送货时间有特定要求或有协助上架、贴标签等要求。
延迟订单处理方式	延迟订单的处理方式，可事先约定规则，避免过多的临时询问或紧急处理。

流程五 输入订单资料

订单分类后，将客户订单、电话、传真等基本信息输入订单处理系统。

在订单录入的过程中容易出现数据录入不及时，易出错。为了更好地提高订单录入的效率，减少差错率，制订以下的流程和操作标准：

（1）对订单录入员进行培训，提高作业速度和录入数据的熟练程度。

（2）根据订单的紧急程度把订单分类处理，把订单处理部门设立一个快件订单处理部门和一个普通订单处理部门。

（3）统一客户订单表格格式、填写要求，特别是传真订单、电话订单、Email 订单，这样便于客户管理和提高效率。

（4）订单分区作业，设立电话、传真订单接收区等和订单录入作业区，减少相互干扰。

如图4-6所示：

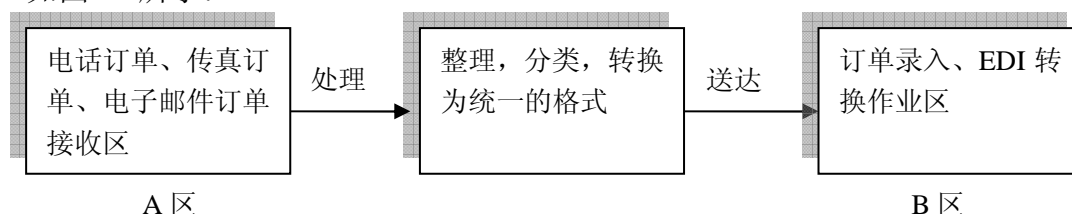


图 4-6 分区作业流程图

安得公司接受的电话订单、传真订单和电子邮件订单先由A区整理好转到B区（订单录入和EDI转换）集中处理。

（5）联机作业（主要是订单录入），提高作业效率。当订单数量非常多的时候，可安排多个订单录入人员联机录入。订单录入时设置一块订单录入作业板，

和电脑屏幕平行放置，用打洞机把集合起来的订单放在板上，便于数据录入，提高作业的效率，方便保存订单文本。

流程六 流程存货查询及订单分配存货

订单资料输入系统并确认无误后，主要的处理作业是如何作有效地汇总、分类、调拨库存，以便后续的物流作业能有效地进行。

1. 存货查询

存货查询是确认有效库存是否能够满足客户需求，库存商品资料一般包括品项名称、库存量、已分配库存、有效存货。

2. 分配库存

库存分配模式可分为单一订单分配及批次分配两种。

(1) 单一订单分配，又称在线即时分配，即在输入订单资料时就将存货分配给该订单。

(2) 批次分配，是指累积汇总已输入订单资料后，再一次分配库存。批次分配的原则和说明如表4-3所示：

表 4-3 批次划分原则说明表

批次划分原则	说明
按接单时序	将整个接单时间划分几个时段，若一天有多个配送时段，将订单按接单先后顺序分为几个批次处理。
按配送区域路径	将同一配送区域路径的订单汇总处理。
按流通加工需求	将有加工或流通加工的需求订单汇总处理。
按车辆需求	若配送商品需特殊配送车辆(如低温车、冷冻车、冷藏车)，可汇总合并处理。

确定批次分配的订单后，若某商品总出货量大于分配的库存量，可根据以下原则分配有限的库存。如表 4-4 所示：

表 4-4 有限库存分配原则说明表

批次划分原则	说明
特殊优先权者分配	缺货补货订单、延迟交货订单、紧急订单或远期订单，应确定这些订单的优先分配权。
根据客户等级，对客户重要程度高的作优先分配	按客户等级分为 ABC 类客户进行库存分配。
根据订单金额作优先分配	对公司贡献度大的订单作优先处理。
客户信用等级作优先分配	信用较好的客户订单作优先处理。
系统定义作优先分配	建立一套订单处理的优先系统。

3. 分配后存货不足的异动处理

如果现有的库存数量无法满足客户需求，且客户又不愿意接受替代品时，则跟据客户意愿与公司政策来决定对应的处理方式。存货不足处理流程如图 4-7 所示：

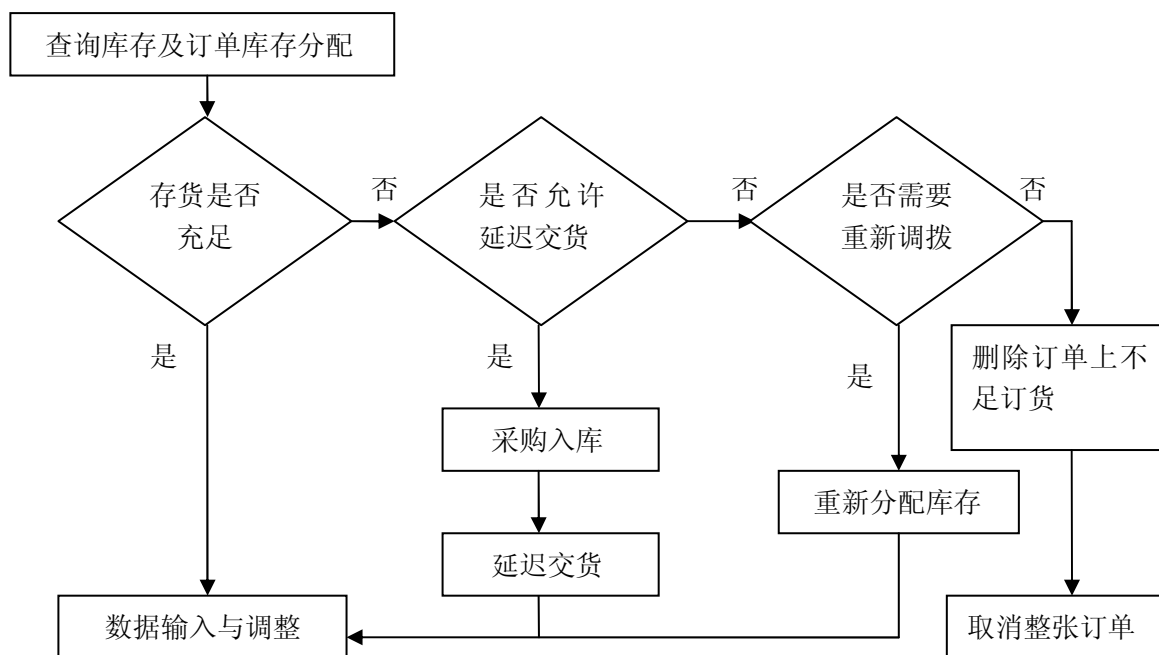


图 4-7 存货不足处理流程图

根据客户的情况类别和公司的实际情况，分配后存货不足的异动处理如表 4-5 所示：

表 4-5 分配后存货不足的异动处理

情况类别	约束条件	处理说明
客户不允许过期交货	公司无法重新调拨	删除订单上不足额的订货，或取消订单。
	公司可以重新调拨	重新调拨分配订单。
客户允许不足额订单	公司不希望分批出货	删除订单上的不足额部分。
客户允许不足额订货补送	等待有货时再予以补送	等待有货时再予以补送。
	处理下一张订单时“补送”	与下一张订单合并配送。
	有时限的延迟交货，并要求一次配送	客户允许一段时间的过期交货，并要求所有订单一次配送。
	无时限的延迟交货，并要求一次配送	等待所有订货到达再出货。
根据公司政策	允许过期分批补货	分批出货的额外成本高，宁可取消客户订单,不愿意分批补货。

流程七 订单数据处理输出

订单处理结果的输出，如拣货单、出货单等，要根据这些输出单据进行出货作业。订单资料处理输出的单据类别如表4-6所示：

表 4-6 单据类别说明表

单据类别	说明
拣货单	拣货单提供商品出库指示，作为拣货的依据。拣货单的格式应配合物流中心的拣货策略及拣货作业方式，以提供有效的拣货信息，便于拣货的进行。拣货单按商品储位顺序打印，以减少人员行走距离。
送货单	<p>(1) 交货时交送货单据给客户清点签收，作为收货凭证。要确保送货单上的资料与实际送货相符，除了出货前的清点外，出货单据打印时间及订单的异动信息非常重要。</p> <p>(2) 单据打印时间，保证送货单与出货单完全相符合的方法是使用手持终端对商品条形码进行扫描，在电脑上修改完毕后，再打印出货单。</p> <p>(3) 送货单资料，送货单据上的资料除了基本的出货信息外，还需注明订单异动情形，如缺货品项或缺货数量等信息。</p>
缺货资料	<p>(1) 库存分配后，对于缺货的商品或缺货的订单信息，系统提供查询或报表功能，以便及时处理。</p> <p>(2) 库存缺货商品，提供按商品别或供应商别查询的缺货商品资料，以提醒采购人员商品缺货。</p> <p>(3) 缺货订单，提供按客户别查询的缺货订单资料，以便对订单作跟踪处理。</p>

订单处理完毕以后，分拣作业人员拿到拣货单，再根据单据的分拣信息进行备货作业。

4.3.2 备货流程的标准化

(一) 备货的标准流程

备货标准化流程如图 4-8 所示：

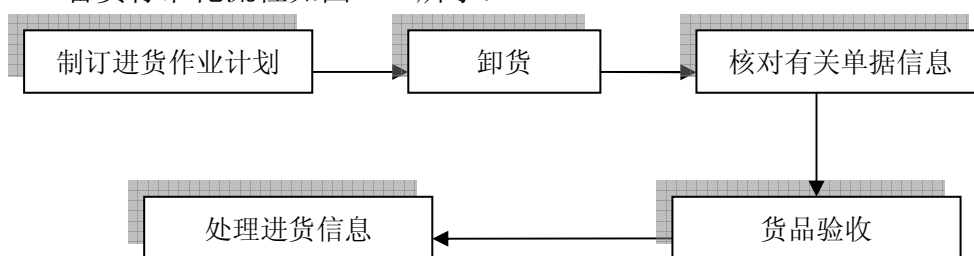


图 4-8 备货作业流程图

备货作业的内容包括：从送货车上将货物卸下，并核对该货物的数量及状态（数量检查、质量检查、开箱等）。

1. 制订进货作业计划

制订进货作业计划的目的是跟据订单所反映的信息，掌握商品到达的时间、品类、数量及到货方式，尽可能准确预测到货时间，作出卸货、储位、人力、物力等方面的计划和安排，保证整个进货流程的顺利进行，同时提高作业效率。

(1) 储位准备

储位准备是根据预计到货的商品特性、体积、质量、数量和到货时间等信息，

结合商品分区、分类和储位管理的要求，预计储位，预先确定商品的理货场所和存储位置。

(2) 设备器材的准备

设备器材的准备是根据到货商品的理化性能及包装、单位重量、单位体积、到货数量等信息，确定检验、计量、卸货与搬运的方法，准备好相应的检验设施、度量衡、卸货及堆货工具与设备，并安排好卸货站台空间。

2. 核对有关单据信息

进货商品通常都有采购订单、采购进货通知，供应方开具的出仓单、发票、发货明细表等单据，进货时应核对货物与单据反映的信息是否相符。

3. 货品验收

货品验收工作包括品质的检验和数量的点收。在商品验收核对时主要是核对商品条形码（或物流条形码）、核对商品的数量、包装上的品名、规格、颜色、等级、细数等。

4. 处理进货信息

商品经验收确认后，再填写“验收单”，并将有关入库信息及时准确地录入库存商品信息管理系统。

(二) 储存标准

储存是按照分拣、配货工序的要求，在理货场地储存少量货物。

货物存放的原则：

(1) 周转快、周转量大和储存待运的商品，或搬运困难的货物，应存放在靠近出口处。

(2) 具备必要的装卸搬运机械，应尽可能充分、合理利用空间。

(3) 根据储存方法和搬运机械的要求规划、设置好通道。

(4) 注意防火，注意防损，如防止掉落或撞击、防潮等。

(5) 商品实施先进先出。

(三) 流通加工流程标准

流通加工标准化流程如图 4-9 所示：

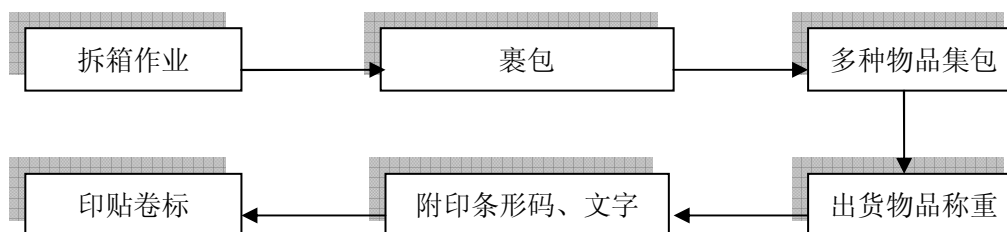


图 4-9 流通加工作业流程图

流通加工是在物品从生产领域向消费领域流动的过程中，为促进销售、保护产品质量和提高物流效率。流通加工和一般的生产型加工在加工方法、加工组织、生产管理方面并无显著区别，但在加工对象、加工程度方面差别较大，在流通加工的设计中应当注意流通加工的合理化和流程的简化。

4.3.3 分拣及配货流程的标准化

(一) 分拣的标准流程

分拣标准化流程如图4-10所示：

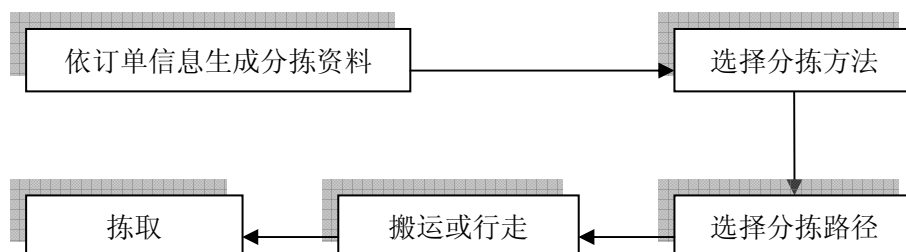


图 4-10 分拣作业流程图

订单处理后，要根据输出的信息对各个客户的订单进行分拣作业。分拣作业是一项复杂、工作量很大的活动，尤其是在用户多，所需品种规格多，而需求批量又较小的情况下，假如客户需求频率很高，就必须在很短时间内，完成分拣配货工作。所以，如何选择分拣配货工艺、如何高效完成分拣配货，在某种程度上决定着安得公司的服务质量和经济效益。

1. 拣货方法

拣货是根据拣货单据拣出顾客需要的货物。拣货的方法根据不同标准主要分为按单和批量分拣，在拣货的方法选择上要遵循以下适用条件。如表 4-7 所示：

表 4-7 拣货作业方法

按单分拣适用情况	批量分拣适用情况
<p>①用户不稳定，波动较大，不能建立相对稳定的用户分货货位，难以建立稳定的分货线。在这种情况下，宜采用灵活机动的拣选式工艺，用户少时或用户很多时都可以采取。</p> <p>②用户之间的共同需求不是主要的，而差异很大，统计用户共同需求，将共同需求一次取出再分给各用户。</p> <p>③用户配送时间要求不一，有紧急的，也有限定时间的。</p> <p>④用户需求的种类太多，增加统计和共同取货的难度。</p>	<p>①连锁企业内部的配送中心，用户都是自营的商店，用户稳定且数量较多。</p> <p>②用户的需求有很强的共同性，货物种类相同，需求差异较小。</p> <p>③用户需求的种类有限，易于统计，且分拣时间不至于太长。</p> <p>④用户对配送时间没有严格要求。</p> <p>⑤适合对效率和作业成本有较高要求的配送中心。</p> <p>⑥专业性强的配送中心，容易形成稳定的用户，货物种类有限，适合采用批量拣选方法。</p>

(二) 配货作业的标准流程

配货作业标准化流程如图4-11所示：

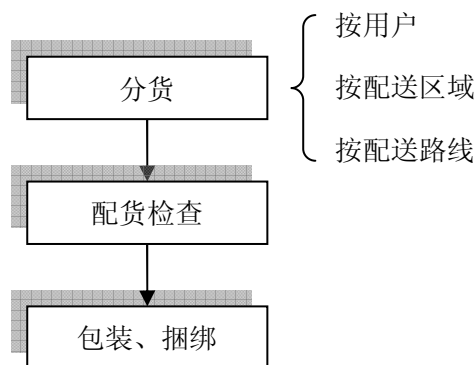


图 4-11 配货作业流程图

完成拣取后的商品按订单的用户、配送区域或配送路线进行分类，再进行配货检查，装入适当的包装或进行捆包，做好标识和贴标签的工作之后，再根据客户和行走路线等指示将物品运至出货准备区，最后装车配送。

1. 分货

分货的方法有按客户分货、按配送路线分货，按配送区域分货三种方式。它们的适用情况如下：

- (1) 当客户的货物满车装载或基本上达到满车装载时，按照用户分货。
- (2) 当处理属于不同客户的小批量多品种货物，且客户所在区域分布集中时，要按照配送区域分货。
- (3) 当处理属于不同客户的小批量多品种货物，且客户所在区域分布不集中时，要按照配送路线分货。

2. 配货检查

进行完分拣后，要把分拣出来的货物集中在分拣区进行配货，配货完毕后要检查核对配货的货品、数量等和用户是否相符。

3. 包装和捆绑

必要时，还要增加一个流通加工的过程，为即将配货装车的货物进行适当的包装、捆绑、贴标签、印刷广告等活动。

4.3.4 配装流程的标准化

(一) 配装作业的标准流程

配送作业的标准化流程如图 4-12 所示：

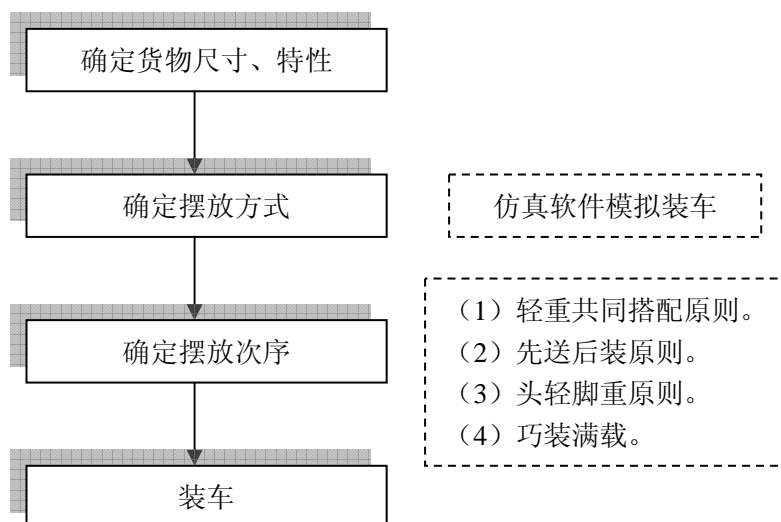


图 4-12 配装作业流程图

1. 确定货物尺寸与性质

为了更好地进行配装作业，在配装前要了解配装货物的重量、保存方式、物理状态、尺寸等性质。要考虑配装的货物是否能承受挤压，货物是常温保存还是低温保存，以及货物相拼是否有安全隐患。

2. 确定货物的摆放方式

(1) 不同货物的形状和不同的摆放方式所得的摆放货物的数量是不相同的，也就是不同的摆放方式对车辆的装载量利用率是不同的。下图所示是几种不同的摆放方式，(如果货物是规则的长方体或正方体)货物的摆放方式如图 4-13 所示：

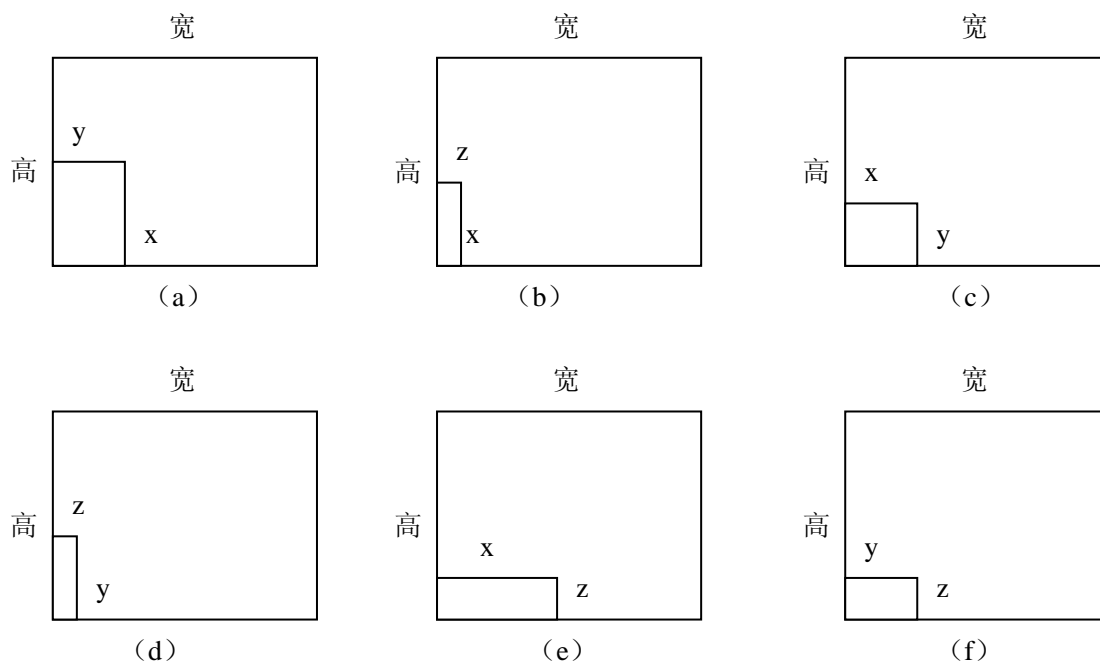


图 4-13 长方体包装货物在车内可能的配置方案

包装的规格和开启方式，开口的设计，按长、宽装车的选择，货物的摆放次序，货物受到的挤压，都影响到运输的质量。利用电脑模拟（仿真软件）对货物进行配装，可以降低人力物力以及配装的时间和难度。

（2）由于大部分的车辆的车厢都是长方体的，所有要根据货物的大小，尽量让缝隙留在表面积最小的车厢面。下面将提供一个配装的实例：

假设车厢长 12 米，宽 2.8 米（内径）高 4.5 米如图 4-14 所示：

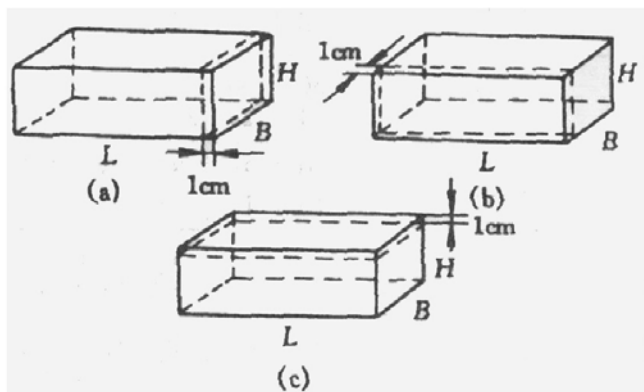


图 4-14 车辆少装 1cm 空置的容积

如图4-14，图(a)空置的体积为 $2.8 \times 4.5 \times 0.01 = 0.126\text{M}^3$ ，空置率为 $0.126/151.2 = 0.083\%$ ，如图(b)空置的体积为 $12 \times 4.5 \times 0.01 = 0.54\text{M}^3$ ，空置率为 $0.54/151.2 = 0.35\%$ ，如图(c)空置的体积为 $12 \times 2.8 \times 0.01 = 0.336\text{M}^3$ ，空置率为 $0.336/151.2 = 0.22\%$ 。

所以对于配装的车辆是长方体时，缝隙的空位应该如图(a)所示，这样可以最大限度地节省空间，提高装载率。

（3）由于重体货物能充分利用车辆载重量，而不能充分利用车辆容积，轻体货物能充分利用车辆的容积，却不能充分利用车辆的载重量，所以采用轻体货物与重体货物配装的方法可以同时最大限度地利用车辆的载重量和容积。同一车内轻重货物的配装量可通过以下公式计算：

$$\begin{cases} P_1 + P_2 = P \\ \frac{P_1}{\gamma_1} + \frac{P_2}{\gamma} = V \end{cases}$$

$$\text{则： } P_1 = (P - V \times \gamma_2) / (1 - \gamma_2 / \gamma_1) = [\gamma_1 \times (P - V \times \gamma_2)] / \gamma_1 - \gamma_2$$

$$P_2 = P - P_1$$

式中：P为车辆标记载重量（吨）；

P_1 为重体货物装载量（吨）；

P_2 为轻体货物装载量（吨）；

V为车辆全部有效容积（立方米）；

γ_1 为重体货物比重（吨/立方米）；

γ_2 为轻体货物比重（吨/立方米）。

3. 运用配装作业原则确定摆放次序

配装作业在实际装车时必须遵循以下标准：

- (1) 轻重共同搭配原则。
- (2) 先送后装、先装后送原则。
- (3) 头轻脚重原则。
- (4) 巧装满载

① 包装箱要装尽可能多的货物；

② 集装箱要装尽可能多的包装箱；

③ 实行解体运输。这针对一些体积大且笨重、不易装卸又容易碰撞致损的货物所采取的一种装载技术。例如，大型机电产品、科学仪器、自行车、缝纫机等，可将其拆卸装车，分别包装，以缩小其所占据的空间位置，达到便利装卸搬运和提高运输装卸效率的目的。随着安得公司业务的不扩展，其业务也涉足到笨重机械产品的运输，如案例中提到的安得公司的I客户就是著名的机械品牌；

④ 实施单元化。改进包装技术，逐步实行单元化、托盘化，可以提高装载量。

利用以下公式算出具体货物的配装重量，可以提高吨位的利用率。

$$W_A + W_B = W$$

$$W_A \times R_A + W_B \times R_B = V$$

其中两种货物的配装重量为 W_A 、 W_B ，质量体积分别为 R_A 、 R_B

$$\text{则 } W_A = \frac{V - W \times R_B}{R_A - R_B} \quad W_B = \frac{V - W \times R_A}{R_B - R_A}$$

安得公司的业务主体一般是快速消费品、建筑材料和家电。在实行拼箱作业时，即多种不同大小的货物进行拼箱时，装载率往往是安得担心的一个因素。以下，以美的家电产品为例，可用上述公式解决拼箱问题。

假设 $R_A = 0.9 \text{ m}^3/\text{T}$ $R_B = 1.6 \text{ m}^3/\text{T}$ $W = 11 \text{ 吨}$ $V = 15 \text{ m}^3$

$$W_A + W_B = 11$$

$$W_A = \frac{15 - 11 \times 1.6}{0.9 - 1.6} = 3.71 \text{ T}$$

$$W_B = \frac{15 - 11 \times 0.9}{1.6 - 0.9} = 7.29 \text{ T}$$

A产品装载3.71T，B产品装载7.29T使得车辆的装载容积充分利用。

4.3.5 送货及结算流程的标准化

送货及结算的标准化流程如图 4-15 所示：

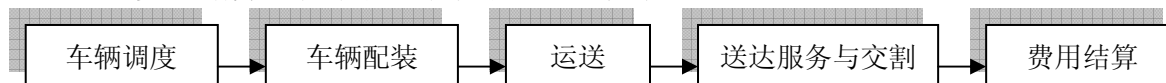


图 4-15 送货作业流程图

(一) 车辆调度

送货人员根据调度人员的送货指示（出车调派单）执行送货作业。当送货人员接到出车指示后，将车辆开到指定的装货地点，然后与出货人员清点分拣配装好的货物，由装卸人员将商品配装上车。

(二) 车辆配装

根据不同配送要求，在选择合适的车辆的基础上对车辆进行配装以达到提高

利用率的目的。在具体装车时，装车顺序或运送批次先后一般按照用户的要求时间先后进行，但对同一车辆共同配送的货物装车则要将货物依“后送先装”的顺序。有时在考虑利用车辆的空间的同时，可能还要根据货物的性质、形状、体积及重量等，做出弹性调整，如轻货应放在重货的上面，包装强度差的应放在包装强度好的上面，易滚动的卷状、桶状货物要垂直摆放等等。

（三）运送

根据配送计划所确定的最优路线，在规定的时间内及时准确地将货物运送到客户的手中，在运送过程中要注意加强运输车辆的考核与管理。

（四）送达服务与交割

当货物送达交货地点后，送货人员应协助收货单位将货品卸下车，放到指定位置，并与收货人员一起清点货物，做好送货完成确认工作（送货签收回单）。如果有退货、调货要求，则应随车带回退调商品，并完成有关单证手续。

（五）费用结算

配送部门的车辆按指定的计划到达客户处并完成配送工作后，即可通知财务部门进行费用结算。

花城分公司在财务结算时常遇到配送回单不完整、逾期率高，批量回单遗失的问题，还有近 40 万的应付未付款。针对这些情况从流程方面的解决方法如下：

1. 确认客户信用。不论是何种订单，接受订单后都要核查客户的财务状况，以确定其是否有能力支付该订单的帐款，通常的做法是检查客户的应收帐款是否已超过其信用额度。其流程如图 4-16 所示：

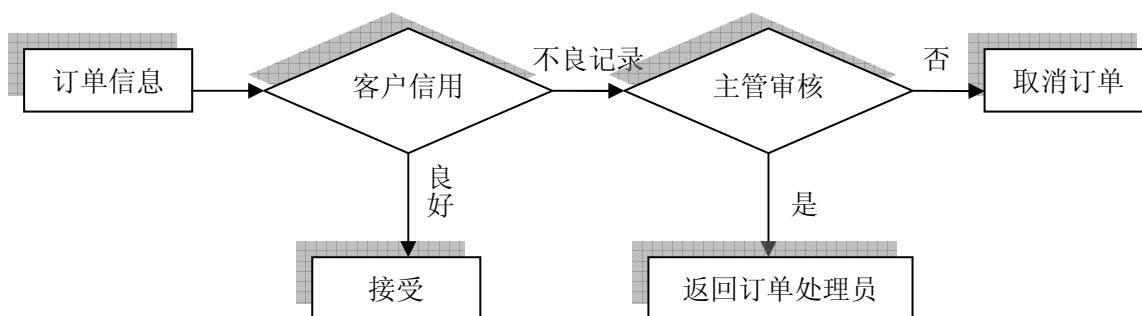


图 4-16 确认客户信用流程图

2. 增加一个回单处理流程。运送司机有一个配送回单的流程，每个司机送货后都要有回单签认工作，作为该司机业务的衡量标志。其流程如图 4-17 所示：

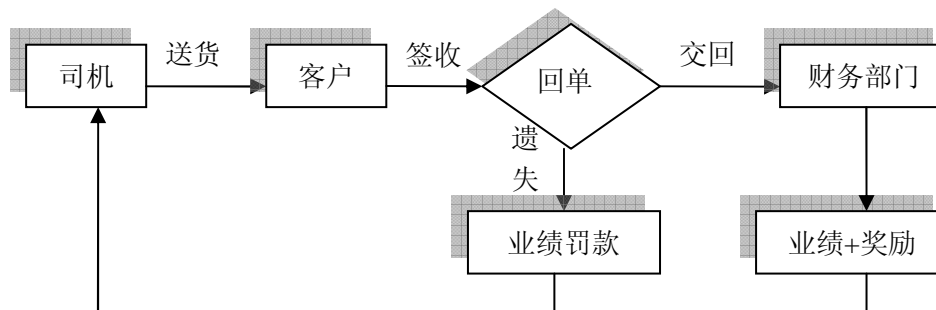


图 4-17 回单处理流程图

上述环节要求司机每送完一次货物都要到财务部门返还送货回单，并以此作为考核司机业绩的一个重要的项目，并且根据司机配送的执行情况，年终时给予奖励。

4.4 安得公司配送问题解决方案

4.4.1 对“P 分公司共同配送”问题提出的解决方案

（一）共同配送问题分析

安得公司由于没有根据实际的运作流程制订标准化的流程，所以造成执行过程中遇到很多问题例如安得公司在共同配送中存在的问题主要是客户单次订单批量太小，而客户计划下达订单时间不统一，每个客户配送区域不统一，因此很多配送路线无法进行集拼以实现共同配送，无法提高集拼率。所以根据具体的情况统一下订单的截至时间，根据配送的具体情况制订共同配送流程的标准化作业，规范业务的操作，提高作业效率和降低作业成本。

（二）用标准化流程提出解决方案

1. 方案简介

安得公司应该根据具体的情况统一下订单的截至时间，确定一个最小的订单批量 and 标准的订单批量。安得的分公司之间或者与其他物流公司建立数据共享平台，实行共同配送，规范业务的操作，提高作业效率和降低作业成本。

2. 共同配送具体运作流程

（1）统一下单时间

为了集中配送，提高集装率，先要统一下单时间。对于定时下订单的措施，实施对策如下：

①预测——在订单下达前根据客户的货物数量与下单习惯进行预测。

②按配送地区划分客户下达订单的时间段，规定其下达订单时间，具体操作可以通过客服部经理与客户磋商。具体的改变如表 4-8 和表 4-9 所示：

表 4-8 原来下订单情况

客户	配送区域	下单时间
A	安徽全省	下单不定时
K	合肥、巢湖以及阜阳的皖中及皖北地区	5 点定时下单
C	合肥、蚌埠、淮北、安庆以及阜阳等地区	下单不定时
D	合肥以及巢湖的部分地区	下单不定时

表 4-9 统一下订单时间后的下订单情况

客户	配送区域	下单时间
A	安徽全省	上午 10 点~中午 12 点
K	合肥、巢湖以及阜阳的皖中及皖北地区	下午 3 点~下午 5 点
C	合肥、蚌埠、淮北、安庆以及阜阳等地区	中午 11 点~下午 1 点
D	合肥以及巢湖的部分地区	下午 5 点~晚上 7 点

③针对统一配送路线上地方的远近进行差异化下单，如下图芜湖——蚌埠——宿州——阜阳——合肥——安庆这一条线路中，芜湖是起点，安庆是最后一个配送目的地，根据配装原则中“后送先装”原理，车辆会最后来到安庆，所以，在装车的结构上，安庆的货物应当放在车厢的最里面。由于装车必须要在订单处

理和配装的前提下进行，因此，这一条线路的最远目的地安庆，应该提早下达定单，其他的配送地点由远及近以此类推。如图 4-18 所示：

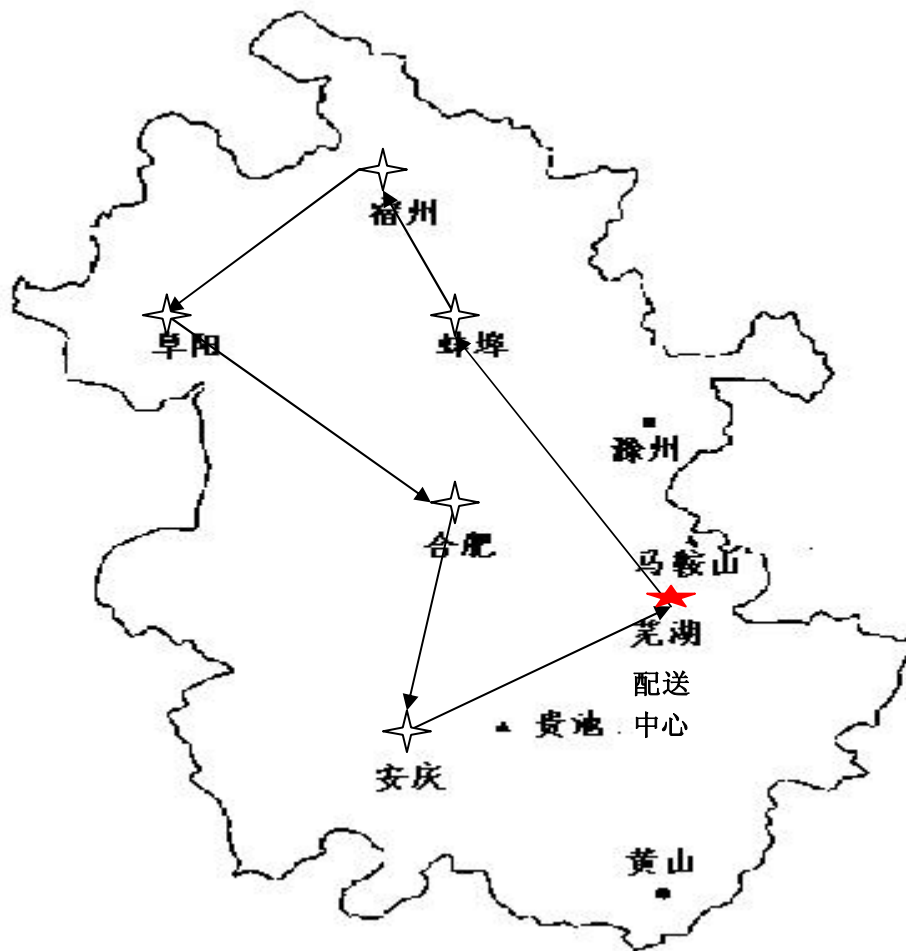


图 4-18 配送路径图

(2) 安得公司应该提供几个标准化的服务，根据这些不同的服务水平收取不同的服务费用，客户可为高效的配送付出较高的价格，以享受更优质的服务，具体如表 4-10 所示：

表 4-10 不同承诺收取不同收费

类型（配送承诺时间）	24 小时内	两天内	3~5 天
价格 (针对同种产品而言)	50 元/吨.公里	46 元/吨.公里	42 元/吨.公里
适用产品	快速消费品如：饮料、食品、部分家电产品	家电产品	建材产品、机械商品

(3) 联合其他物流企业，在安得分公司和其他物流公司之间建立数据共享平台，扩大拼箱货源，开发新的客户，加强彼此的业务联系与合作。

① 分公司之间的合作

假设广州的 T 分公司 3 月 20 日有 600 箱的电饭煲（约 10 吨），运输单价假设为 200 元每吨，运给河北的 A 客户（运到芜湖的单价为 100 元每吨）。在芜湖的 P 分公司在 3 月 21 日左右也有 500 箱的压力锅（约 10 吨），运输单价为 90 元每吨（运量达到规模如 20 吨以上 80 元每吨），运给河北的 B 客户。

具体流程如图 4-19 所示：

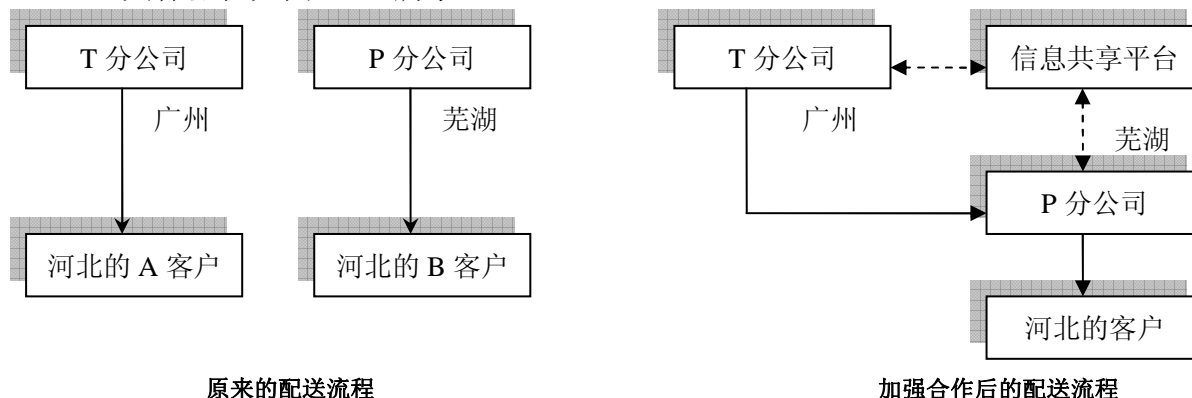


图 4-19 配送流程改进图

流程说明：原来的配送流程是各个分公司之间的作业完全独立的没有信息的共享和业务的合作。加强合作后的配送流程，由于建立的共享的信息平台，所以原来各自独立地向河北的客户配送的总成本（ $200 \times 10 + 90 \times 100 = 2900$ ）2900 元，加强合作后的配送总成本为（ $100 \times 10 + 20 \times 80 = 2600$ ）2600 元，节省了 300 元。

②企业（物流公司）的合作

零担货物的共同配送流程如图 4-20 所示：

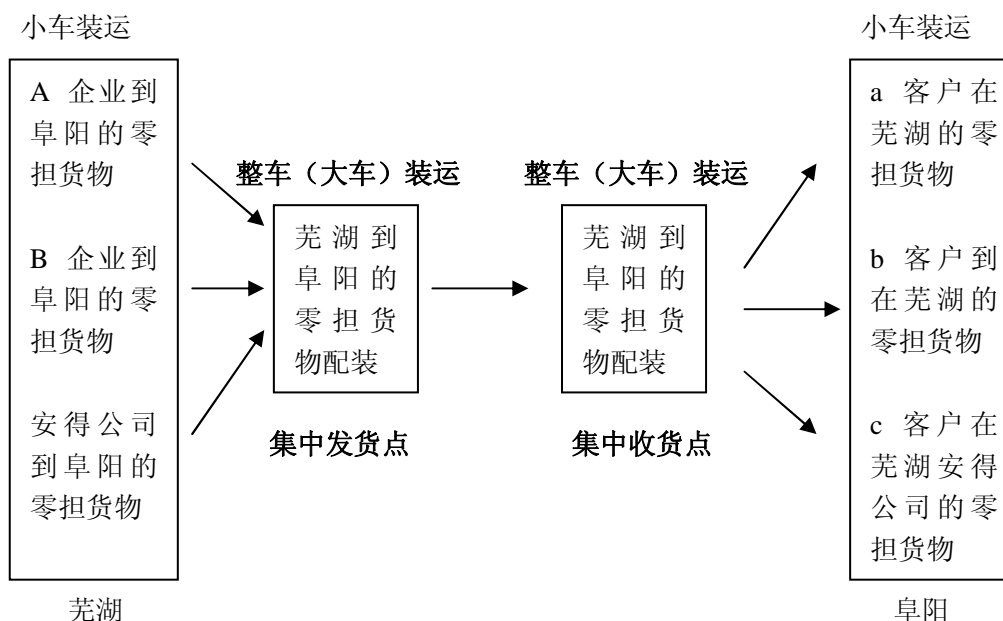


图 4-20 零担货物的共同配送流程图

流程说明：安得公司应该加强与其他企业（物流公司）之间的合作，努力建立一个共享的信息平台，充分利用共享信息，把零担货物整合成整车或者是批量的运输，降低单位的成本。从芜湖各个零担的货物用小运到当地集中的发货点，拼装成大车后，再运送到阜阳的集中收货点然后再转小车配送到具体的客户，实现零担货物的共同配送，降低零担货物的配送成本。

(4) 扩大合作路线，签订合作路线范围和与更多稳定的合作伙伴加强合作。

配送到乡镇地区的作业流程如图 4-21 所示：

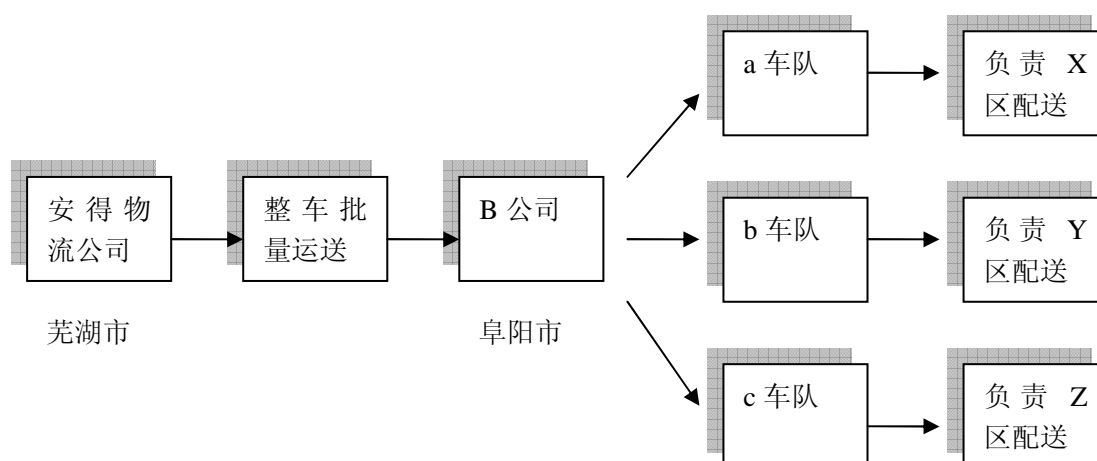


图 4-21 配送到乡镇作业流程图

流程说明：大车转小车运输，大批量运送货物到某区域集中，然后根据各地区的需求与当地的车队签订合作协议，明确各个地区的配送负责车队。对于安得公司业务的配送区域要覆盖到乡镇，而且批量太小，配送车辆调度困难。进一步扩大开发乡镇级的市场，对于偏远的地区可采取这种作业流程。

(5) 共同配送根据不同对象实施

如家电、建筑材料、日用品等快销品属于恒温保管范畴，实施共同配送比较简单。若配送对象如乳业、食品、饮品等快销品，属于冰冻和未冷保管范畴的物品配送，在共同配送选择合作伙伴时一般选择相同行业的客户一起配送。

(6) 确定一个最小的标准订单批量，使客户便于按单位（如一个托盘）、卡车或货车计算。

(7) 鼓励措施，如：饮水机的订单批量在 100 台以内按原价配送，100～500 台优惠 5% 配送，500 台以上优惠 10% 配送。

4.4.2 对“配送增值服务”问题提出的解决方案

(一) 配送增值服务问题分析

随着公司城市配送、终端用户配送的业务不断优化与深入拓展，配送业务模式和服务需求从原来的 B2B 式配送在渐渐向 B2C 模式转变。然而，简单的从仓库到客户手中的物流服务已不能满足客户的期望,安得企业通过争取第一时间掌握与了解终端市场，提供更高层次的物流服务，以此来体现出其竞争力和自身价值。客户的新的需要主要有一下几点：

1. 售后服务方面的特殊要求：电器产品的调试、安装、保修等服务；
2. 安全性的特殊要求：货损率低、丢失、受潮、事故；
3. 成本性的特殊要求：低成本；
4. 时间的特殊要求：准时，准点；
5. 实时信息的掌握：货物的在途情况查询等。

安得公司送货的司机或搬运工人在没有相关的专业知识或赶时间的情况下，无法帮客户进行安装或调试。由于缺少规范的流程和作业标准，在配送过程中，司机或送货人员无法为顾客安装、调试产品的现象经常都发生，有的顾客还以此

为由拒绝签收。

（二）用标准化流程提出解决方案

1. 方案简介

安得公司在作业流程上应该制订司机送货流程中安装调试和售后服务处理的标准化的作业流程，规范员工的管理，及时地满足顾客提出的新的需求，争取利用标准化的作业提高服务水平，进而争取更多的客户。

2. 配送增值服务具体运作流程

（1）送货上门及安装调试，作业流程如图 4-22 所示：

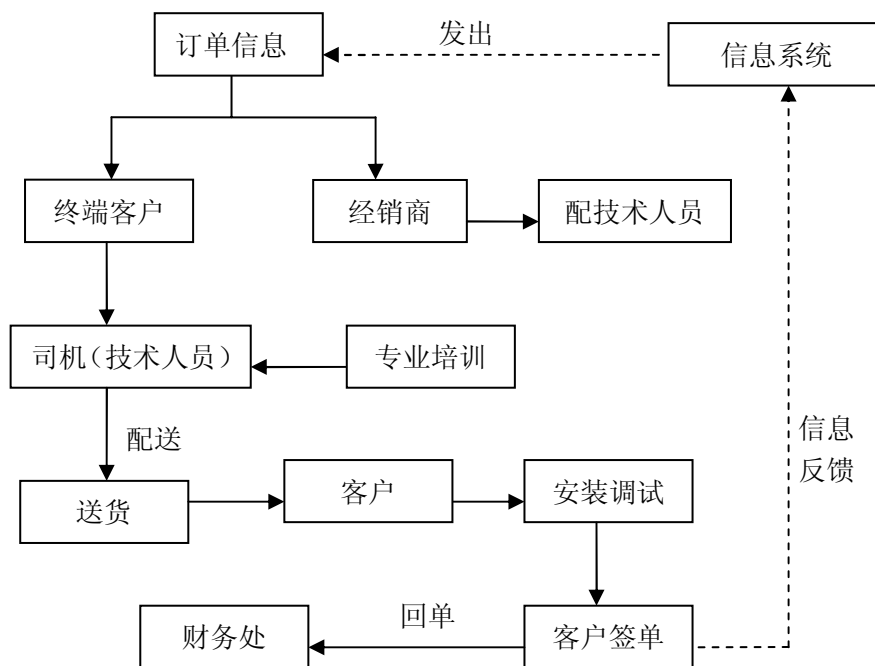


图 4-22 调试安装作业流程图

流程说明：根据订单的信息，分为终端客户和经销商的配送。对于经销商要根据具体的情况配技术人员（一般情况都不需要）。对于终端客户配送的司机或送货人员要进行专业的培训，不仅能让让他们胜任配送和装卸搬运，而且可以为终端客户解答商品的一些常见的疑问和进行相应的调试和安装。司机或送货人员配送到终端客户的流程规定，每个司机或送货人员都有专门为客户安装和调试的时间，在配送的计划中已经把安装和调试的时间考虑了（一般都为 30 分钟，具体要根据不同产品的要求安装调试的时间适当延长）。司机或送货人员在安装调试完后，要客户签单确认，然后把回单送回财务处备案，同时要把终端客户的信息及时反馈。

（2）售后服务处理主要针对终端客户，其标准化流程如图 4-23 所示：

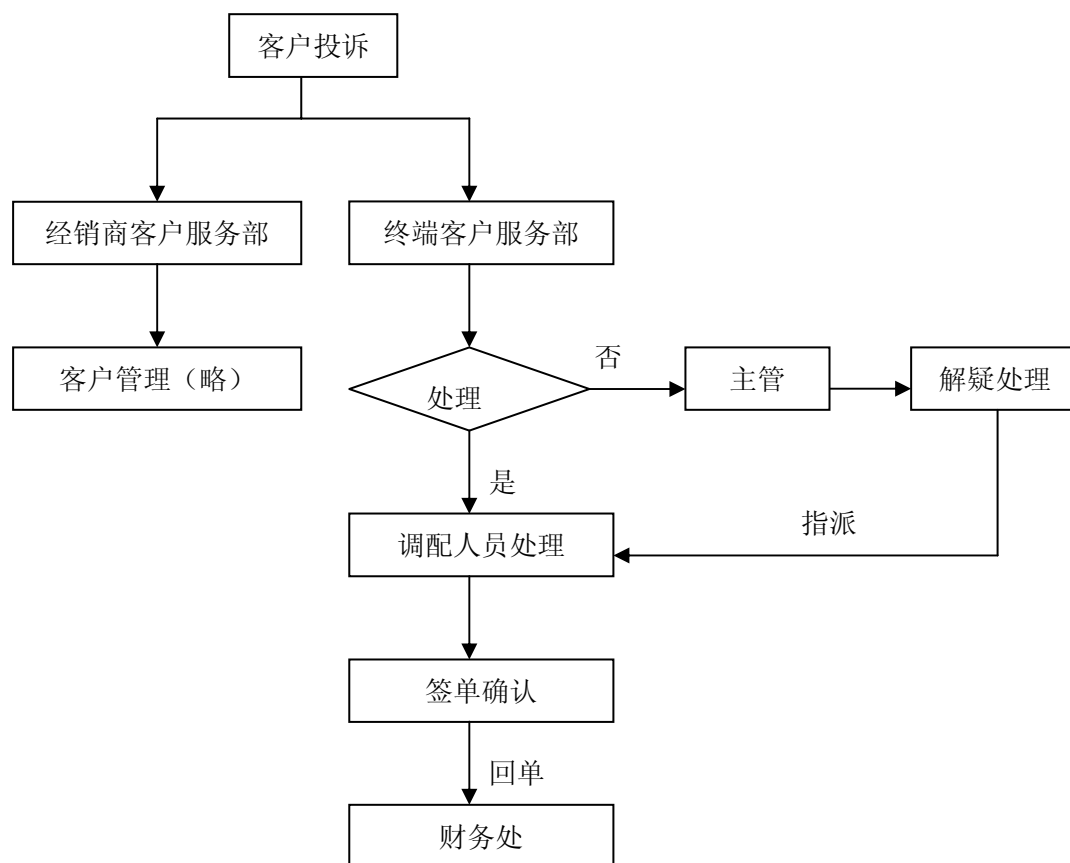


图 4-23 售后服务处理流程

流程说明：根据安得公司具体的不同业务，把安得公司的客户服务部分为经销商客户服务部和终端客户服务部。经销商客户服务部专门解决经销商，零售商等客户问题投诉的部门。终端客户服务部是根据近几年终端客户不断增多的情况而设立的专门解决终端消费者问题投诉的部门。终端客户服务部门收到客户的问题投诉，由专门的客户服务人员处理，如果客户服务人员无法解决的问题就交由部门的主管处理，部门主管会根据实际的情况为客户解答疑问，最终指派作业人员处理。客户服务人员亲自上门服务，为客户解决完问题后，由客户签单确认，回单送到财务处备案。

（3）配送过程实时的信息情况的掌握。客户要求更多的参与配送过程，及时了解货物的配送情况，信息系统是满足客户这种需求的有力保证。

（4）精益配送。快速消费品要求准时、准点配送，安得公司的业务也涉及到快速消费品，所以实行精益配送可以适应客户的这种需要。

（5）黄昏配送。上班族（城市白领阶层）白天上班忙，没有时间，在工作时间送货时没有人在家，只有下班后才有空闲时间，所以针对这些特殊的顾客群采用特殊的配送方法，满足顾客新的需求。

4.4.3 对“淡旺季作业不均”问题提出的解决方案

（一）淡旺季作业不均问题分析

安得公司业务的大部分都是家电产品，客户的销售旺季分别出现在元旦、春

节前、五一、十一，而对于那些季节性较明显的产品（如空调、电风扇、冰箱）也会随着销售季节的到来而增加大量的订单。大量的订单在同一时间处理，作业量猛增，无论是人力或者设备都会出现严重的不足，再加上作业的流程不规范，没有紧急的处理流程，造成管理难度加大，管理混乱。而在淡季的时候出现人员的冗余或车辆的闲置，这不但增加的公司的成本支出同时也给公司的形象造成一定的影响，降低员工的工作激情。所以制订一个针对淡旺季作业人员 and 车辆的操作流程的标准将使管理更加规范，有章可循，操作更加简明，轻松应对突发的需求和季节性的不均衡需求。

（二）用标准化流程提出解决方案

1. 方案简介

制订一个针对淡旺季作业人员 and 车辆的操作流程的标准将使管理更加规范，有章可循，操作更加简明，轻松应对突发的需求和季节性的不均衡需求。

2. 淡旺季作业具体运作流程

（1）处理旺季作业人员 and 车辆不足的标准化流程如图 4-24 所示：

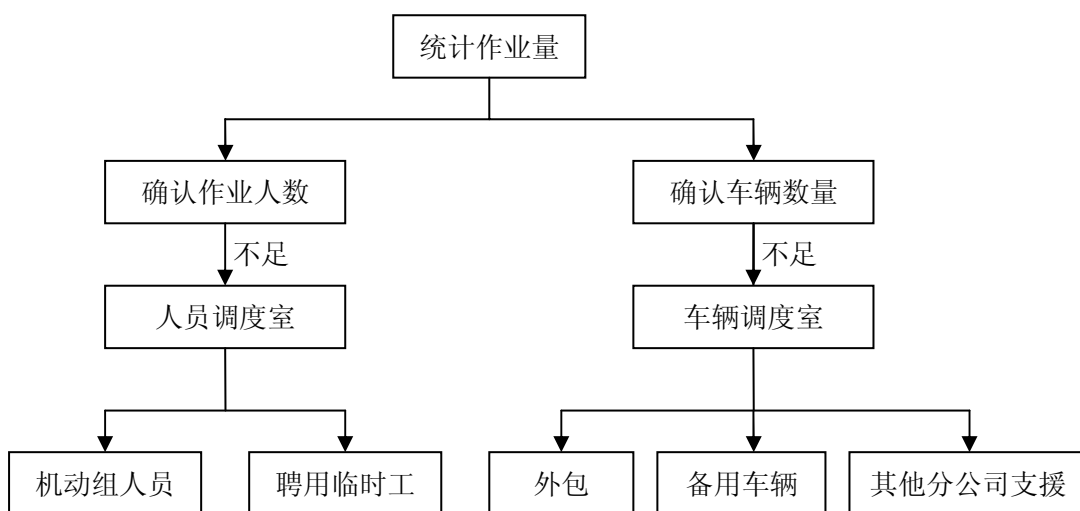


图 4-24 作业人员不足处理流程

流程说明：预测或根据订单信息统计作业量，确认作业的人数和车辆的数量。当作业人员不足时，人员调度室可根据实际的情况，调用机动组（主要是日常的管理人员或其他部门的人员或保安等）或聘用临时工，聘用的临时工主要安排在经过短暂的培训即可的作业。当车辆数量不足时，车辆调度室可根据实际的情况，调用备用车辆，或者把业务外包（业务外包时要求业务员跟踪管理，外包的供应商选择主要是平时有业务往来或有一定信誉度的企业或车队），或者请求其他分公司的援助。

（2）处理淡季作业人员 and 车辆过多的标准化流程如图 4-25 所示：

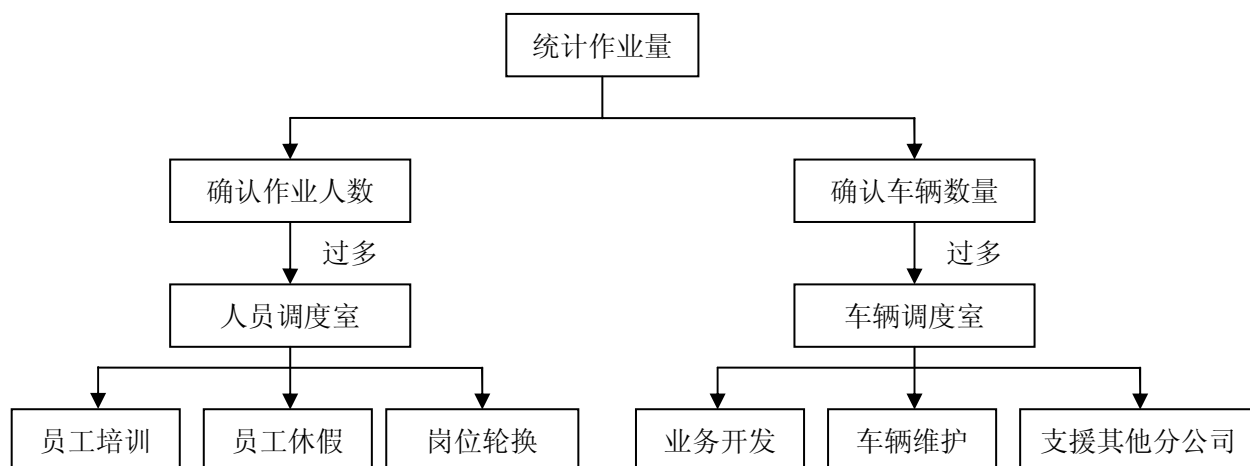


图 4-25 作业人员不足处理流程

流程说明：预测或根据订单的信息统计作业量，确定作业的人数和车辆的数量。当作业人员过多时，人员调度室可安排员工进行培训，提高员工的素质和培养员工的企业文化。也可以安排部分员工休假，组织员工集体旅游，增强员工的归属感和幸福感。同时也可以进行岗位的轮换，让不同的员工临时的岗位轮换，熟悉相关的作业流程，当出现某个作业成为了作业瓶颈时可调用其他部门的员工，由于有岗位的轮换使他们知道流程如何操作，可临时减缓作业量的压力，提高总体的作业能力。当车辆的数量过多时，车辆调度室可根据实际的情况进行业务的开发（充分利用车辆的使用率），对车辆进行维护，或者支援其他分公司。

5 运输流程标准化设计

5.1 安得公司运输现状分析

安得公司具有很强的运输能力，可调车辆上万辆，能够为客户提供优质的运输服务。但是，其运输仍然存在许多问题。

将安得公司运输问题用鱼骨分析法分析如图 5-1 所示：

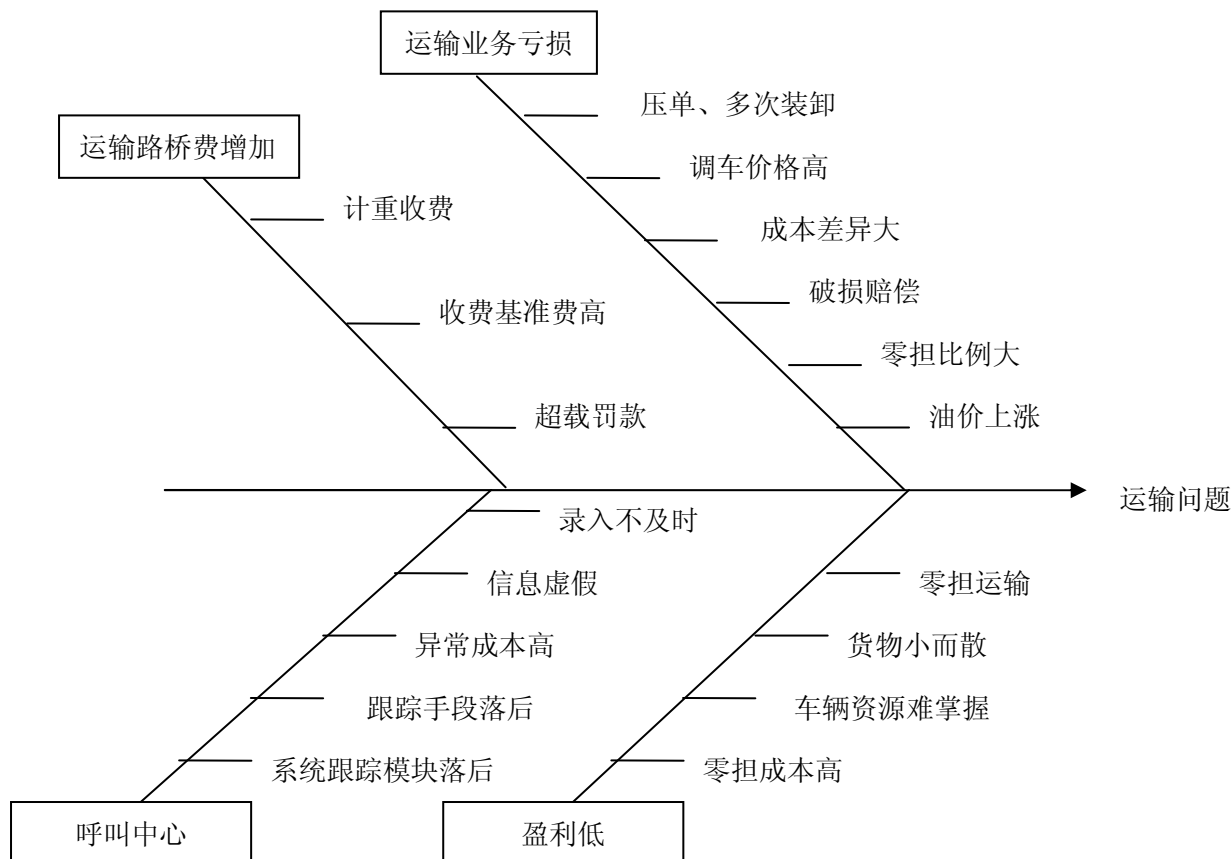


图 5-1 运输问题鱼骨分析图

安得公司虽然长期可调车辆上万辆，但是公司自由车辆十分有限，基本上都是靠外包的车辆进行运输，由于在车辆调度管理不够规范，运输成本控制不力，运力浪费严重。车辆调度的作业流程不规范，在装卸时经常出现堵车现象，导致出货作业效率低。由于车辆在运行途中缺乏有力地监控，出现客户投诉和司机在途说谎话，而且由于缺少司机规范标准的管理，在司机的作业流程上经常出现问题，如回单遗失，信息反馈不及时，信息虚假等，影响了公司的信誉和控制管理的效率。

5.2 运输流程标准化设计概述

5.2.1 运输流程标准化设计的目标

制订一个标准化的运输管理流程，规范司机的管理，明确各自的职责，有效地监控运输全过程，特别是降低运输过程中的异常成本，减少管理费用和提高运输的作业效率。

5.2.2 运输流程标准化设计的基本内容

根据安得公司的运输问题鱼骨分析图，我们总结出运输的运作流程中需要注意的环节主要包括：

1. 车辆调度流程的标准化；
2. 车辆配装流程的标准化；
3. 在途监控流程的标准化；
4. 送达服务交割与结算流程的标准化。

针对运输流程中的这些环节，我们制订标准化的运作，并结合案例对这些运作流程的标准化进行具体说明，在运输流程中，我们充分利用了物流敏捷车辆调度模型，并根据模型在Visual C#环境下编写了程序。通过这个程序，使得复杂的车辆调度决策变成科学的且易于操作的标准化流程，进而能更有效地协助解决其他运输问题。

5.3 运输流程标准化具体设计

运输管理的流程图如图 5-2 所示：

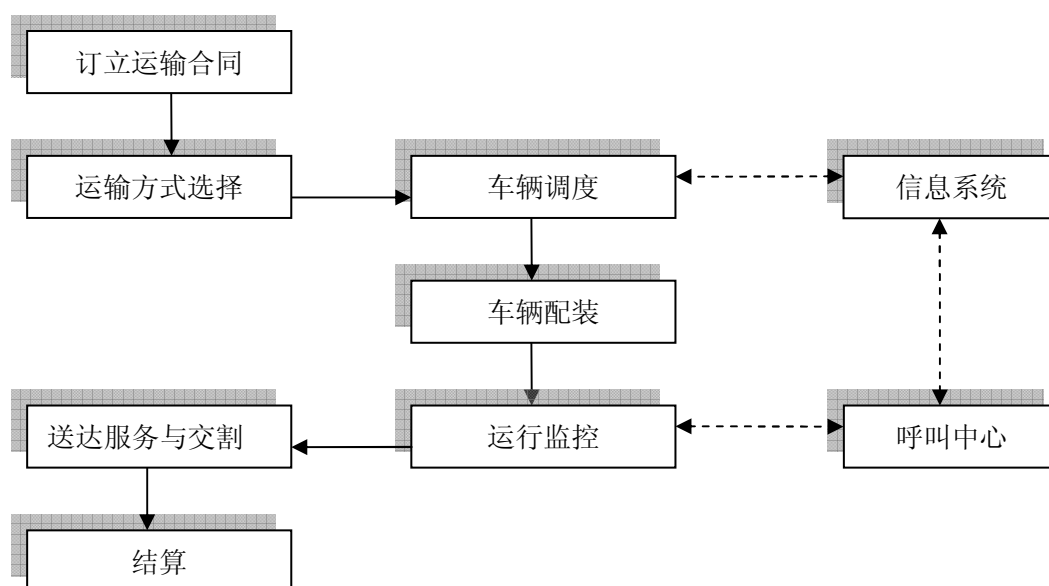


图 5-2 运输管理流程图

经过订单处理后，订单处理员把运输资料发送给车辆调度中心，这样，作业就进入了运输流程阶段。运输流程主要包括与运输队订立合同，运输方式选择，车辆调度，车辆的在途监控，送达服务与交割和最后的费用结算。

5.3.1 受理托运和订立运输合同流程的标准化

安得公司的运输业务种类较多，包括主要的家电产品运输，建材运输，钢板运输等。因此，公司在受理托运并订立合同的过程中，要根据具体业务的需要，和发货人订立有效的合同，与发货人建立长期的合作伙伴关系。下面我们将受理托运和订立合同的过程进行标准化的操作，使管理人员在订立合同的过程中作业规范和有条不紊。

（一）受理托运

受理托运是进行运输的首要工作，安得公司在签订物流运输合同前要和客户办理托运的手续，其工作流程如图 5-3 所示：

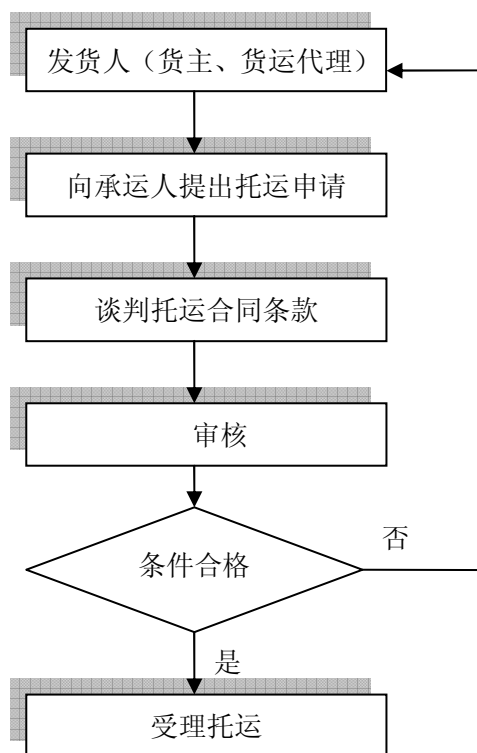


图 5-3 受理托运流程图

如上图所示，这个过程主要是委托方编制托运资料然后向承运人提出托运申请的过程，申请提交后，承运人和托运人审核双方条件，如果条件合适便可以认作托运货物被受理了。

（二）订立运输合同

受理托运之后就是要订立运输合同，为接下来的工作提供法律依据。在公路货运业务中，运输合同主要是运单，运单的签发是运输合同成立的体现。订立运输合同流程如图 5-4 所示：

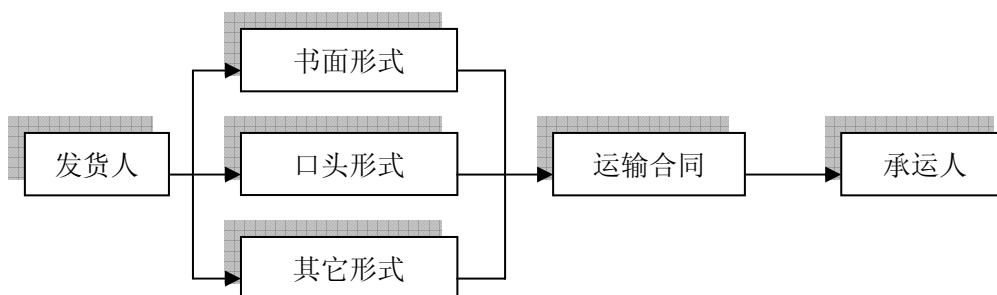


图 5-4 运输合同订立流程图

运输合同的形式一般有书面形式、口头形式和其他形式。订立运输合同一般都是由发货人以书面的形式和承运人签订运输合同。

运输合同的内容如表 5-1 所示：

表 5-1 运输合同内容

运输合同	主要内容
	运单的签发日期和地点
	发货人的名称和地址
	承运人的名称和地址
	货物接管地点、日期一级制订的交货地点
	收货人的名称和地址
	货物品名和包装方法，如属危险产品应说明其基本性质
	货物件数、特征标志和号码
	货物毛重以及其他表示的量化指标
	与运输有关的费用（运费、附加费、关税和其他费用）
	是否允许转运的说明
	发货人负责支付的费用
	货物的价值
	法活人关于货物保险给予承运人的指示
	交付承运人的单据清单
	运输的起止期限

运输合同应该在双方正式签字方能生效，但是对于短途、小批量货物运输以及承托双方具有长期合作伙伴的货物运输，由于双方已经对运输过程中的权利和义务作商定了合同，只需要双方的都认可的说明即可。比如美的集团承运给安得公司的家电运输业务，双方已经订立了长期的合作伙伴关系，不需要每一次承运都订立合同，减少作业的流程。

（三）合同的变更与解除流程

合同在履行过程中如果遇到特殊的情况使合同无法继续履行，这时可以提出变更或解除合同，其运行过程如图 5-5 所示：

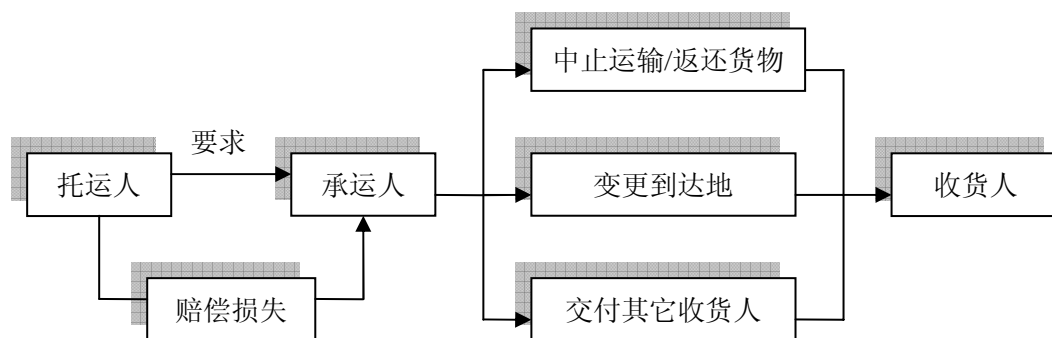


图 5-5 合同变更与解除流程图

遇到特殊情况时，托运人可以要求承运人中止对货物的运输并返还货物，或者选择更改货物的交付地点，又或者更改收货人。对于合同变更的情况，其允许更改情况如表 5-2 所示：

表 5-2 允许变更或者解除合同情况表

允许变更 或者解除合 同情况	不可抗力使运输合同无法履行
	合同当事人一方的原因，在合同约定的期限内确实无法履行运输合同
	合同当事人违约，使合同的履行成为不可能或者不必要
	经合同当事人双方协商同意解除或者变更运输合同（承运人提出解除运输合同的，应退还已受的运费）

合同的变更会影响到双方的利益，在签订或更改合同的时候，应该对合同的条款仔细考虑避免造成利益上的损失。

5.3.2 运输方式选择的标准化

选择合适的运输方式，是公司进行运输规划的重要流程。运输部门在进行运输类型和运输方式的选择时，需要根据公司自己的运营能力和外部条件来决定，无论是选择自营还是外包，都要在满足客户需求的前提下，获得自己的收益。下面我们将安得公司在运输过程中如何选择运输方式进行标准化的操作。

运输方式的选择可以使安得公司更合理地掌握车辆的运行情况和可以调度车辆的数目。运输业务的营运方式包括：自营、外包和协同运输，其中自营包括直接运输和中转运输。选择方式如图 5-6 所示：

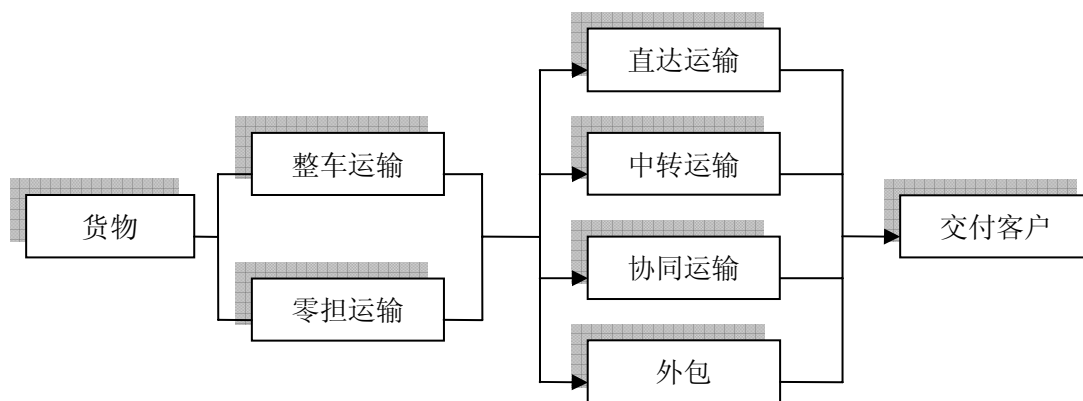


图 5-6 运输方式选择图

(一) 自营运输业务流程

自营运输（直达运输和中转运输）中运输方式的选择流程如图 5-7 所示：

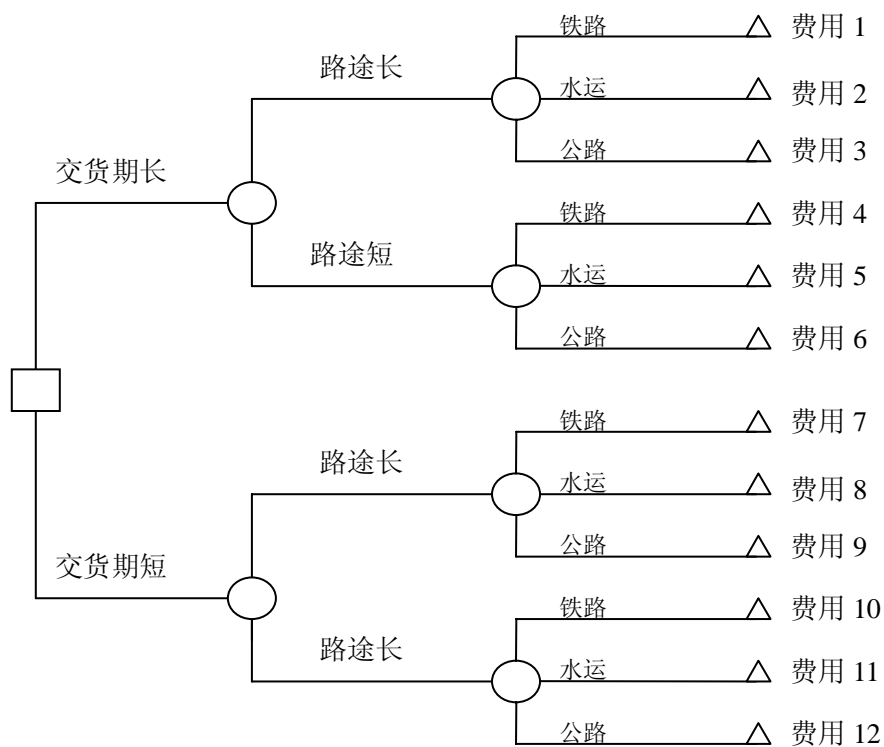


图 5-7 运输方式选择决策树

运输方式是根据交货期限、路程，水、陆路情况和具体费用四个因素逐级决策生成。首先要考虑交货期限的长短，然后考虑路程的长短还有可利用的交通方式，最后考虑费用作出决定选择哪种运输方式。

(1) 铁路/公路

采用铁路、公路运输方式的运作流程如图 5-8 所示：

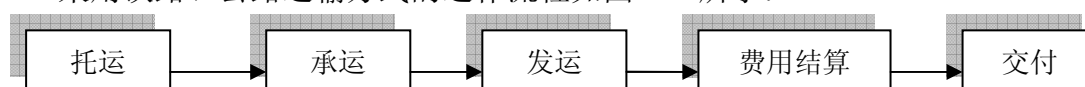


图 5-8 铁路、公路货物运输流程图

(2) 水运

采用水路运输方式的运作流程如图 5-9 所示：

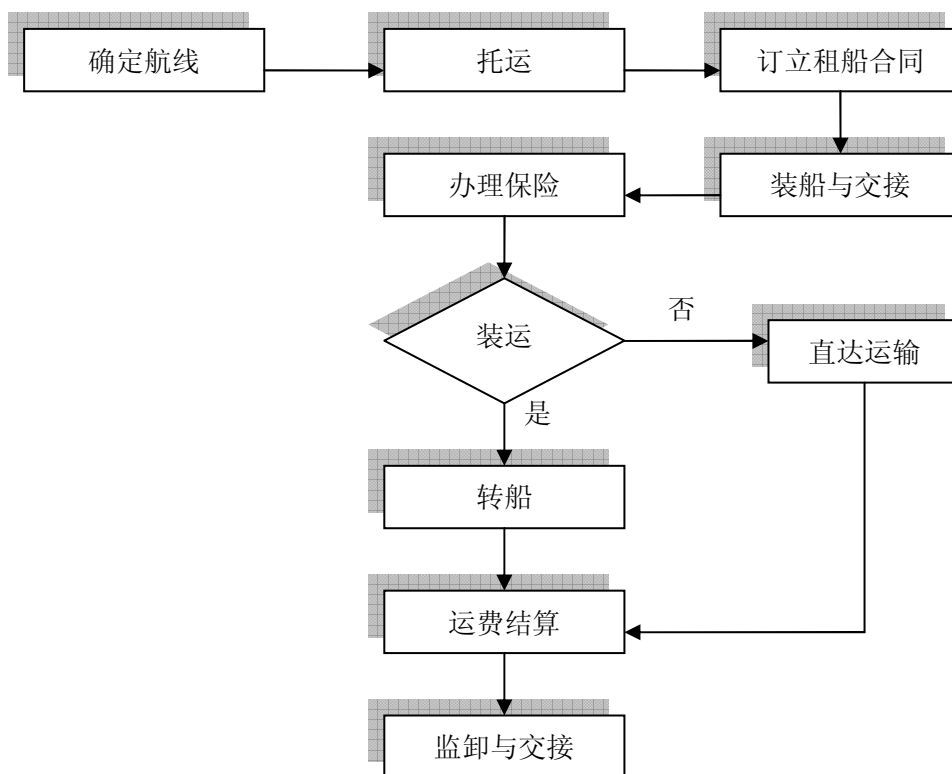


图 5-9 水路运输流程图

在公司的运行过程中，由于特殊情况的要求，运行的方式的选择要结合实际情况，如安得公司在广州到河北的运输中，可以采用海运和中转的方式来满足客户的需求，在进行运输方式流程设计中要考虑各种特殊运输方式，使设计的流程符合运输的需要。

(二) 协同运输作业流程

协同运输在现在物流作业中越来越受人们的关注，协同运输能够有效整合公司和外部的资源，既可以发挥公司的优势，同时可以通过利用外部资源，弥补公司的不足。安得公司应该借此发展协同运输，为客户提供更加优质的服务。

协同运输的建立有效的信息系统，安得公司在这方面有很大的优势，通过安得物流信息系统（ALIS）整合各参加成员，有效控制整个运输的运作过程。协同运输的开展运作流程如图 5-10 所示：

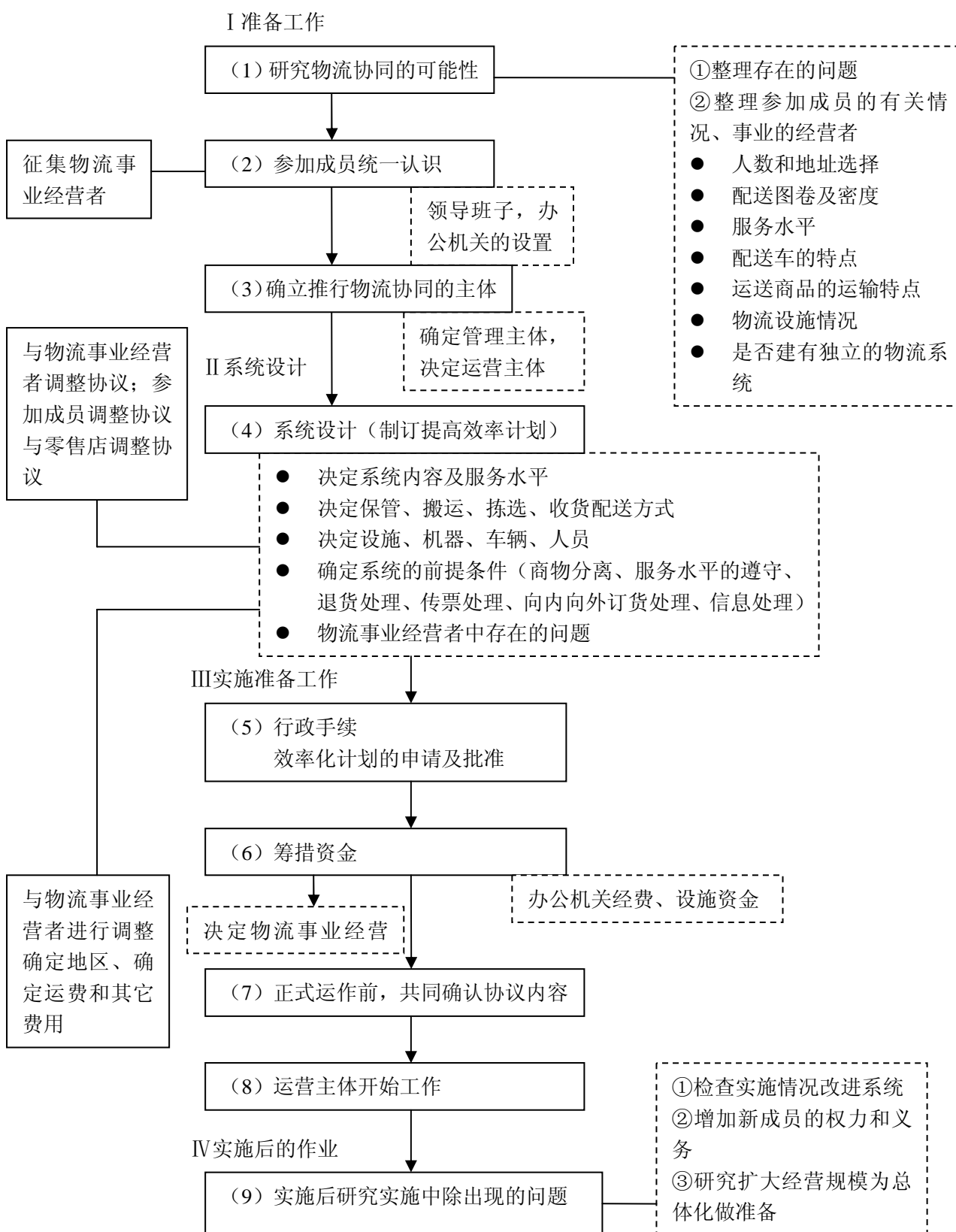


图 5-10 开展协同运输的运作流程图

(1) 协同运输的优势

通过协同运输，可以提高运输的效率，通过统一的收验货，将各自运输，多批次小批量的运输进行整合，从而达到提高运输服务水平的目的。

(2) 协同运输的类型

协同运输由于主体的不同，可分为如下几种类型，如表 5-3 所示：

表 5-3 协同运输类型

类型		显着特征与用途
① 与客户的协同运输		用于采购零部件或采办原材料的车辆均可用于协作运输
② 不同行业货主的协同运输		不跑空车，让物流子公司与其它行业合作，装载回程货或其它公司进行往返运输
③ 集团系统内部的协同运输		企业集团、大资本集团、零售商集团等内部的协同运输
④ 同行业货主的协同运输	与客户的协同运输	组成集团相互合作，共同利用配送储存设施，在运输能力不足的地区互相扶持
	共同出资组建新公司进行协同运输	建立协同物流公司或协同配货公司
	通过同行业 VAN 增值网进行协同运输	以同行业 VAN 增值网为基础进行运输
⑤ 公司主体型	运送者的协同运输	向特定交货点运送货物，交货业务合作化
	共同出资组建新公司开展协同运输	本地的运送公司（特别零担货物运送业者、包租业者）共同出资组建新公司开展送货到户业务
⑥ 合作型	运送业者组成合作机构开展协同运输	运送公司组成合作机构，将各成员在各自收集货物或运输货物地区所收集的货物运到收配货据点，统一运输
	运送合作机构和批发合作机构合作，开展协同运输	运送业者的合作机构和批发商的合作机构合作，设置收集货物和配送货物的据点，运送公司的合作机构统一承包批发商的集货和配货的业务

(三) 运输外包作业流程

物流公司运输业务外包是为了整合企业外部运输资源，从而降低成本、提高效率，发挥自身核心竞争力，专注于自身核心的主要的功能或业务。运输外包的作业流程如图 5-11 所示：

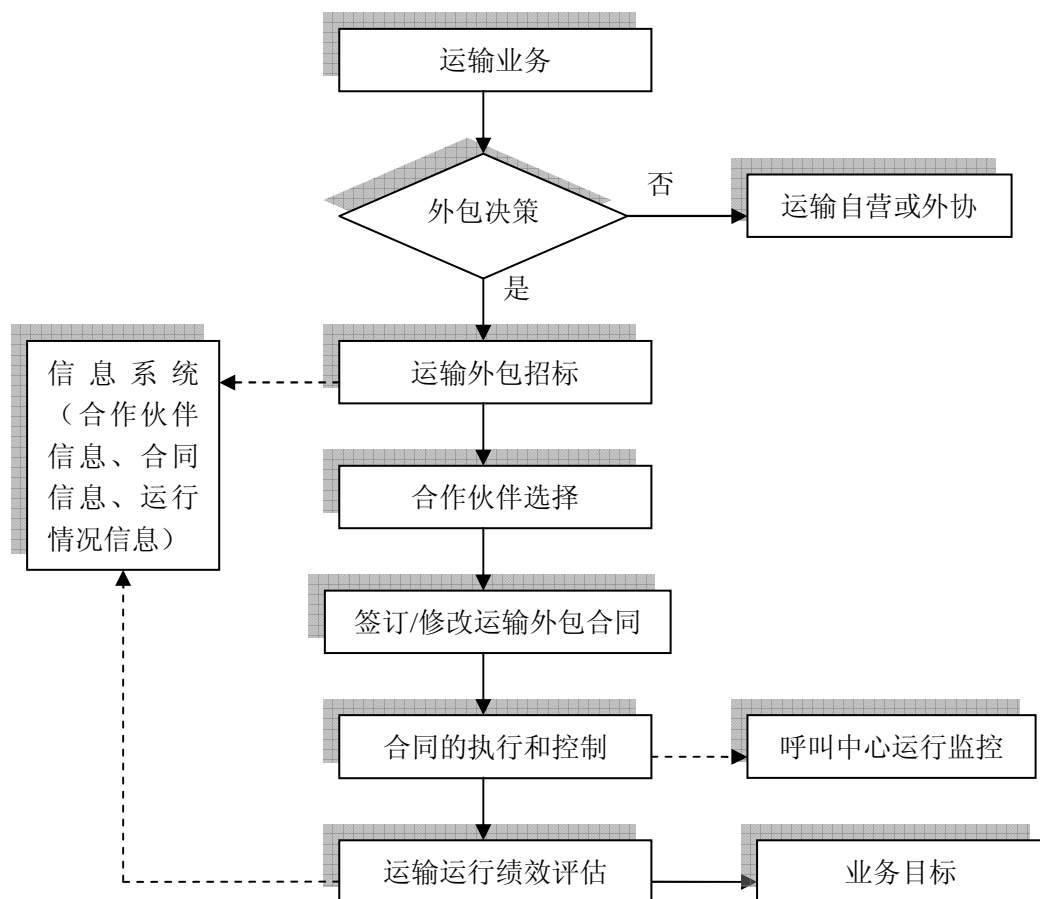


图 5-11 运输外包运作流程

运输外包业务中，合作伙伴的选择尤其重要，与合作伙伴建立战略联盟关系，通过合同的有效管理机制，使双方在运作的过程中达到双赢并建立稳定的基础。

5.3.3 车辆调度流程的标准化

选择了合适的运输方式，接下来就可以针对运输方式进行车辆调度，车辆调度主要包括：运输线路选择，运输工具的选择，运输时间的安排，人员的安排等。安得公司主要进行公路运输，遍布全国的运输网络，为客户提供快捷及时的运输服务。车辆的合理调度能够充分利用车辆的装载率，走最短的路线，在交货时间内把货物送达客户的手中。下面我们将根据安得公司的公路运输业务设计车辆调度流程的标准化。

（一）车辆调度流程标准

1. 车辆调度运作标准流程

车辆调度运作标准流程如图 5-12 所示：

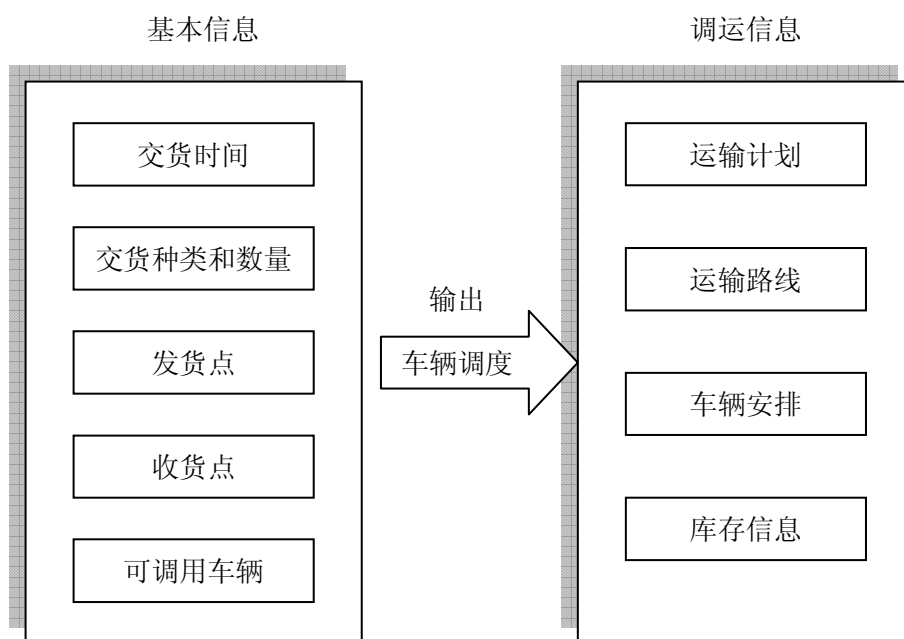


图 5-12 车辆调度流程

安得公司接收运输合同之后，根据合同的要求确定交货的时间、地点、货物的种类数量和收货人的具体信息，通过合理科学的方式进行车辆调度，得出运输计划、运输线路、车辆安排和库存的信息，并为下一次的运输做好准备。

案例中提出，安得公司在进行车辆调度时，主要是依靠管理者的经验、以往的数据、个人的偏好及其对车辆的了解情况、过去工作单位的运作及文化背景做出决策进行调配，这使得调度过程中难以避免主观因素的影响，使决策结果发生误差。

假使使用科学的、量化的方法计算出最优路线，可以减低车辆调度流程中的个人主观因素影响，同时可以减少人力资本的投入，使流程运作更加快捷准确。为了避免这种误差，我们建立了车辆调度优化模型，使凭借经验的车辆调度转化成科学的标准化操作过程。

2. 车辆调度模型与程序的建立

(1) 物流敏捷调动优先技术的提出

基于运输网络和调运优先准则的物流敏捷调运技术的提出，是物流管理信息不断更新和实际运输业务的需求。车辆调度模块能够成为安得物流信息系统（ALIS）的一个有效组成部分，与其他信息模块对接后，实现信息的流通并为公司车辆的调度提供科学的依据。有如下特点：

- ①建立规范的配送网络，即可行运输路径的寻找；
- ②建立基于调运优先准则的递阶控制机制；
- ③基础信息的录入，包括：配送中心的货物库存信息；配送点的货物需求量；可行运输路径的信息；配送中心车辆可调运情况。

(2) 物流敏捷调动优先技术的基础信息描述

① 配送中心货物库存信息

配送中心（或调运中心，Depot）的货物库存信息由以下三元数组 A 表示：

$$A = (i, g_i, q_i)$$

其中, i 表示货物编码, $i=1, 2, 3, \dots$, 简记为 $i \in I$; g_i 表示货物 i 的规格(单位重量、体积等), 并假定所有货物换算为同一规格; q_i 货物 i 的库存数量。

②配送点需求信息

配送点在 t 时刻的实时需求信息由需求矩阵 $C(t)$ 来表示

$$C(t) = (c_{ij}(t)) \quad i \in I, j \in J$$

其中, j 表示配送点编码, $j=1, 2, 3, \dots$, 简记为 $j \in J$; $c_{ij}(t)$ 为 t 时刻配送点 j 对货物 i 的需求量, t 即交货期。

③配送网络(可行运输路径)信息

配送中心至配送点 j 的配送网络的信息由以下三元数组 $R(j)$ 表示

$$R(j) = (r_j, V(r_j), D(r_j))$$

其中, r_j 表示配送中心至配送点 j 的可行运输路径编码, $r_j=1, 2, 3, \dots$, 简记为 $r_j \in R_j$;

$V(r_j)$ 表示路径 r_j 上全部配送点的集合, 且 $V(r_j) = \{j \mid \text{若配送点 } j \text{ 位于可行运输路径 } r_j \text{ 之上, } j \in J\}$;

$D(r_j)$ 表示路径 r_j 上的配送点到配送中心的距离集, 且 $D(r_j) = \{d(v|r_j) \mid v \in V(r_j)\}$,

其中, $d(v|r_j)$ 表示在路径 r_j 上配送点 v 到配送中心的距离, 显然它不一定是配送点 v 到配送中心的最短距离。至于任意配送点 v 到配送中心的最短距离则为

$$D(v) = \min\{d(v|r_v) \mid r_v \in R_v\}$$

基于以上描述的方法, 配送网络所包含的信息, 因可行路径搜索深度和广度而异。这里提出的基于配送网络和调运优先准则的物流敏捷调运技术的基本前提, 就是假定配送中心至其服务区域内所有配送点的可行配送路径已经优化。

④配送中心运输资源(车辆)信息

配送中心的运输车辆信息有以下三元数组 E 表示

$$E = (p, u_p, w_p)$$

其中, p 表示车辆编码, $p=1, 2, 3, \dots$, 简记为 $p \in P$; u_p 表示车辆容量, 并假定其已经换算为与 g_i 同一量纲; w_p 表示车辆运行状态, $w_p \in \{0, 1\}$, 且 $w_p = 1$ 表示车辆处于可调运状态; 反之, 则表示车辆处于正在运输状态。

(3) 调运优先准则

“调运优先准则”是在物流调运中根据交货期迟早、距离远近、运输量大小以及客户重要性程度等信息而对当前全部客户的物流调运优先次序进行的分类、排序。根据调运优先准则设定配送点的当前调运优先级, 进一步可以构成物流调运的递阶控制机制。调运优先准则以实现物流调运敏捷化为策略层面的首要准则, 并按调运策略的层次顺序设置:

第一级调运优先准则: 交货需求早的配送地要优先安排;

第二级调运优先准则: 在交货期相同的配送地中, 距配送中心远的优先安排;

第三级调运优先准则: 针对确定的配送地, 配送车辆吨位大的优先安排;

第四级调运优先准则: 针对确定车辆, 装载物品质量大的优先安排地方。

(4) 物流敏捷调运优化技术的工作逻辑

物流敏捷调运优先技术的工作逻辑由五个工作环节组成, 如图 5-13 所示:

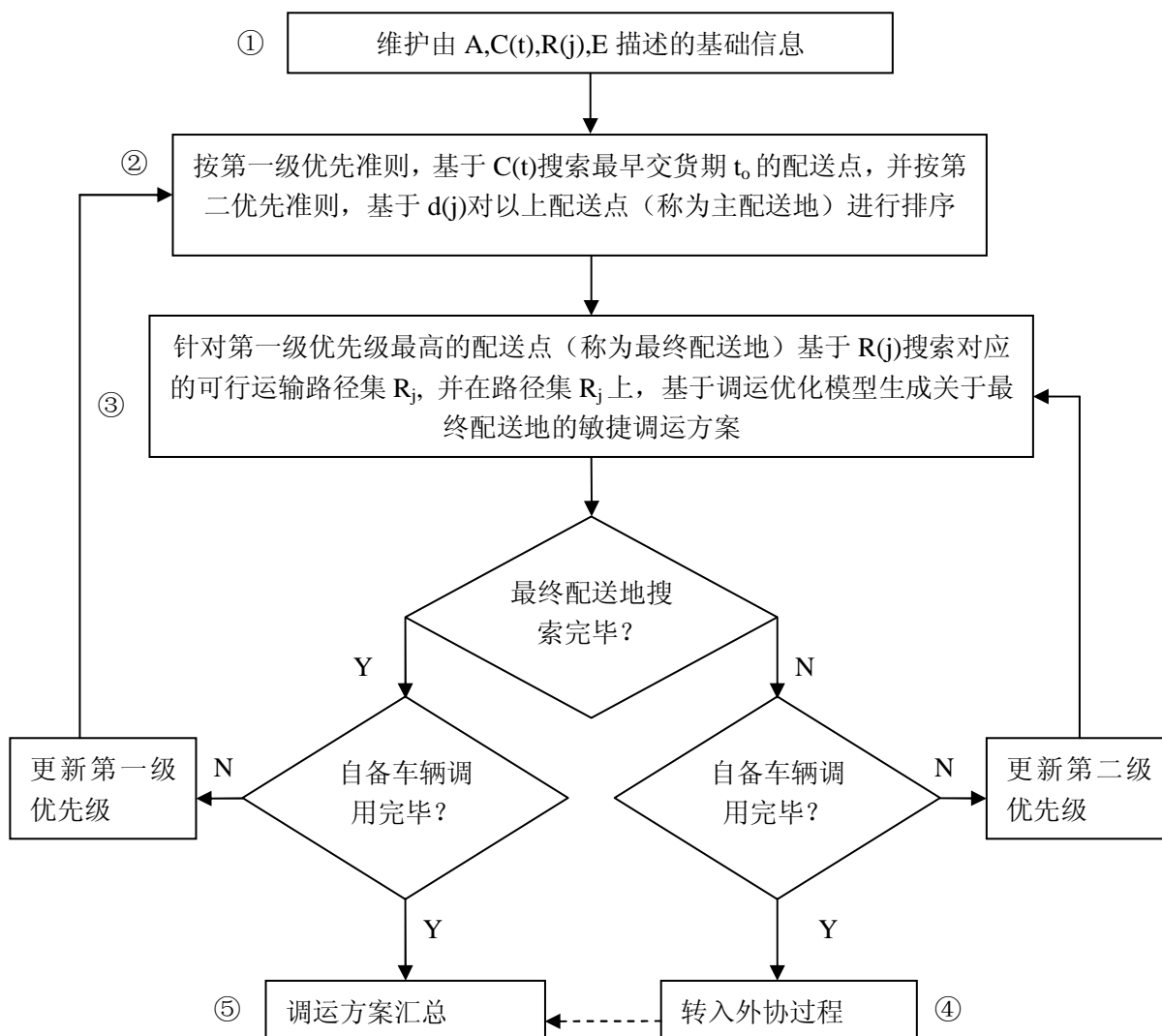


图 5-13 物流敏捷调动优先技术工作逻辑

从图可以看出，“最终配送地”是在最早交货期 t_o 有交货需求且距离配送中心最远的配送点，而且是动态产生的。而实现物流敏捷调动的核心在于第三个工作环节：基于调运优化模型生成关于“最终配送地”的敏捷调运优化方案。

物流敏捷调动优先技术的假设和计算如附录一（第 83 页），物流敏捷调动优先技术在 Visual C# 环境下的编程和操作说明如附录二（第 89 页）。

（5）物流敏捷调运优化系统流程标准化

物流敏捷调动优先技术的提出使物流公司在车辆调运这一环节不再只是依靠经验，而是通过科学的方法进行调运。物流敏捷调运优化系统输入流程包括：

- ① 输入方案的选择（测试已有方案或者自行录入方案）；
- ② 输入配送点个数（包括配送中心，并默认为第一个点）；
- ③ 输入各个配送点之间的距离（无通径的配送点用相对大整数表示）；
- ④ 输入配送中心货物的种类数目；
- ⑤ 输入配送中心各个货物的库存量；
- ⑥ 输入各个配送点需求货物的种类数目和每种货物的交货时间（默认格式如：2007/3/15）；
- ⑦ 输入在交货日期下配送点需求的各种货物的数量；

- ⑧ 输入可调运的车辆数目和各车辆的载重量；
 - ⑨ 基础数据输入结束。
- 操作流程如图 5-14 所示：

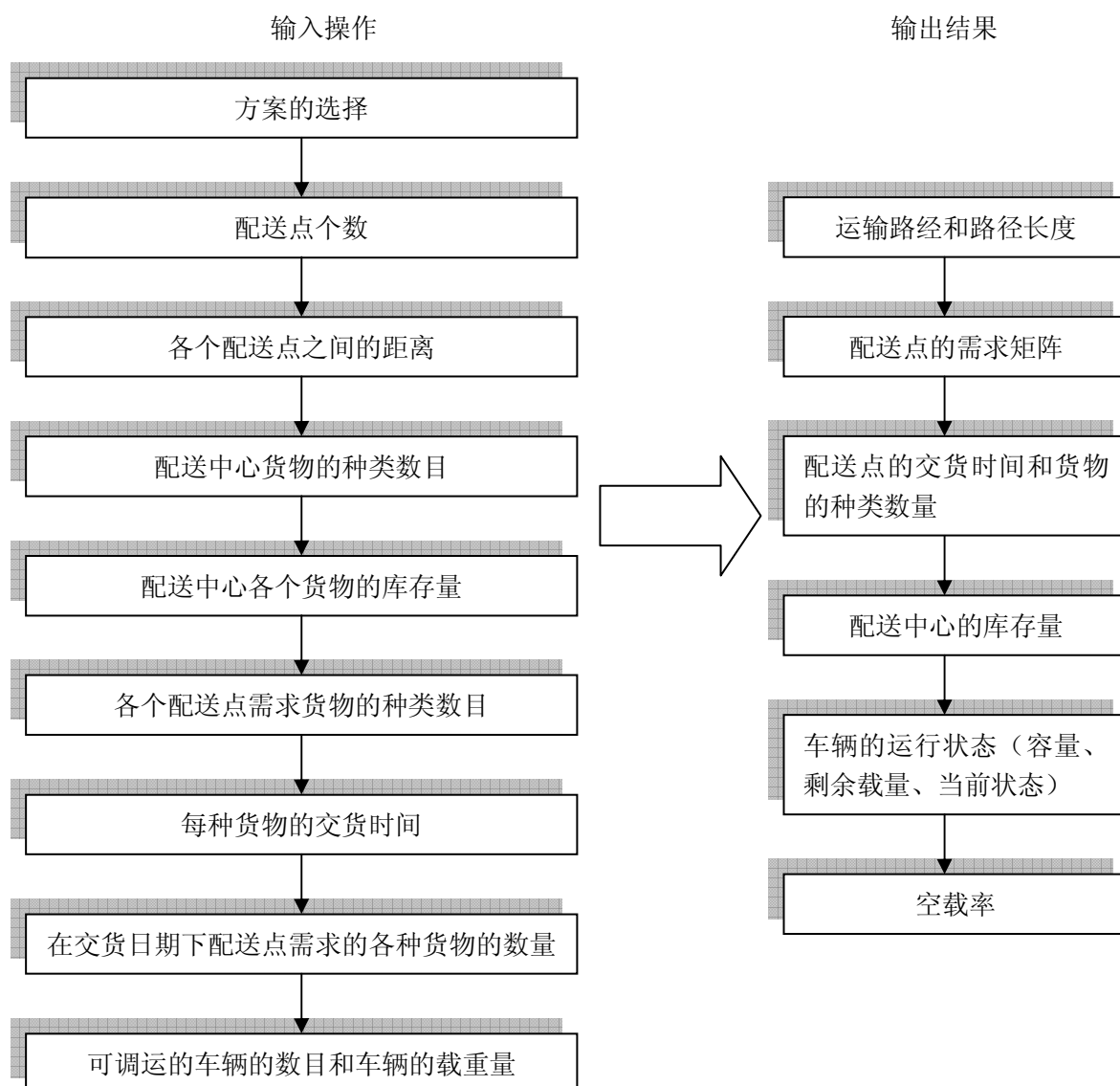


图 5-14 物流敏捷调运优化系统流程图

物流公司进行车辆调运的要求可以概括为：发运及时、空载率小、运费省、仓储占用小等。公司要实现这些目标光靠经验不能满足要求，物流敏捷调运优化系统为实现这些目标提供了有利的条件。为了证明系统的可行性，附录里将会就安得公司案例中的一个例子进行验证。

(6) 车辆调度模型与程序

目前，在安得公司的车辆调度作业主要是由调度员根据经验来完成的。针对此种情况，我们为安得公司编写的这个车辆调度程序是在车辆调度模型的基础上开发的。使用这个车辆调度程序，将会使安得公司的车辆调度流程简化并使作业效率得到提高。

现在，这里将简要介绍车辆调度程序的功能：

① 计算最优路线功能

车辆调度员只需把配送点的数量，配送点两两之间的距离（若两点间不通行则输入大数 M）输入程序内，程序会自动输出最优路线。如：Vs→Vf→Vh（表示最有路线是从 s 地到 f 地再到 h 地）。

② 计算选用车型功能

把货物数量输入程序，可以算出运输成本最小的合适车型。

③ 安排出车日期

通过输入货物需求矩阵，程序可以自动生成出货日期。

④ 计算车辆的剩余载量

程序在安排配送的同时，不仅计算出选用车型，还可以附上车辆装载后的剩余载量。

调度模型与程序的假设条件、特点、操作过程以及更多的功能我们将会在附录里具体说明。运用这个车辆调度程序，将会使得安得公司的车辆调度工作化繁为简，从人工阶段向智能阶段过渡。

5.3.4 车辆配装流程的标准化

车辆装配为车辆的高效运行提供前提，车辆的合理配装能够有效利用车辆的容量，提高车辆的装载量，并保护货物，减少破损率。特别是特殊货物的配装需要更加科学，起到减少意外事故的目的。安得公司要把握好车辆配装这一关，对于长途运输的车辆，通过科学的配装能够提高车辆的利用率，减少破损率，整体减少运输成本。

车辆配装的考虑因素如图 5-15 所示：

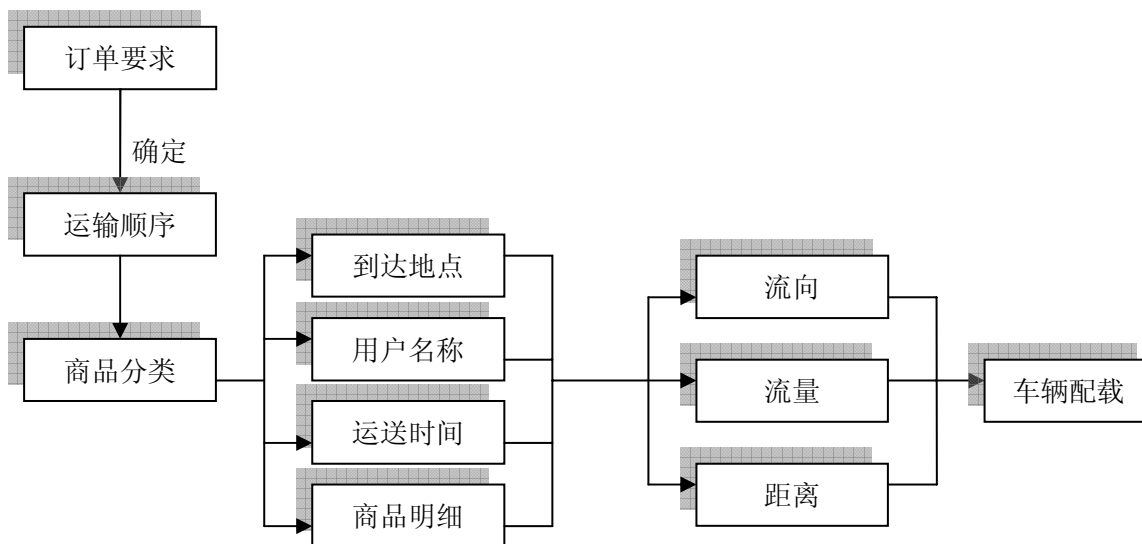


图 5-15 车辆装配考虑因素

根据不同运输要求，在选择合适的车辆的基础上对车辆进行配装以达到提高利用率，是配装的一项主要工作。

车辆配装运作如图 5-16 所示：

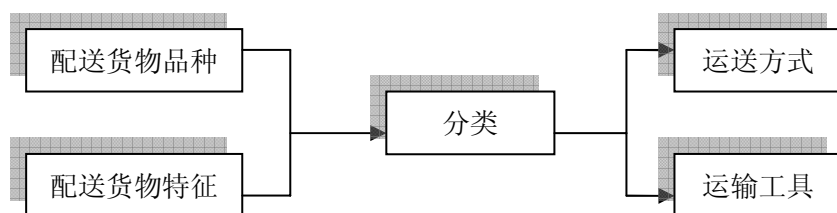


图 5-16 货物配装流程

(1) 由于配送货物的品种和特征各异，为提高配送效率，确保货物的质量，首先必须对特征差异大的货物进行分类，并分别确定不同的运送方式和运输工具。

(2) 特别注意：

- ① 散发臭味的货物不能与具有吸臭性的食物混装；
- ② 散发粉尘的货物不能和清洁的货物混装；
- ③ 渗水货物不能与易受潮货物一同存放；
- ④ 减少或者避免差错，尽量把外观相近、容易混淆的货物分开装载。

(3) 装货的顺序：

① 于配送货物有轻重缓急之分，所以必须初步确定哪些货物可配同一辆车，哪些货物不能配于同一辆车，以做好车辆的初步调配工作；

② 在具体装车时，装车顺序货运送批次先后一般按用户的要求时间先后进行，但是对同一车辆共同配送的货物装车则要将货物依“后送先装”的顺序；

③在考虑有效利用车辆的的空间的同时，根据货物的性质（怕震、怕压、怕撞、怕湿）、形状、体积及重量等做出弹性调整；

④ 放在重货的上面，包装强度差的应放在包装强度好的上面，易滚动的卷状、桶状货物要垂直摆放等等；

⑤ 应按照货物的性质、形状、重量、体积等来具体决定货物的装卸方法。

5.3.5 在途监控流程的标准化

(一) 在途监控作业流程

对车辆进行在途监控，能够确认车辆的具体位置，保证高水平的客服质量。作业流程如图 5-17 所示：

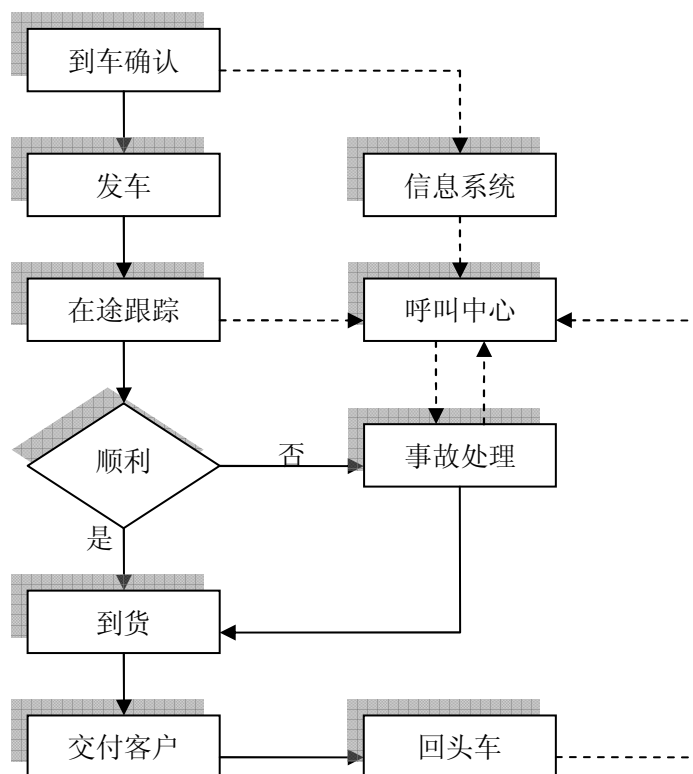


图 5-17 运行监控流程图

根据车辆调度所确定的最优路线，在规定的时间内，司机要及时准确地将货物运送到客户手中，同时，在运送过程中，呼叫中心要加强对运输车辆的考核与管理。

（二）“呼叫中心”作业流程

“呼叫中心”运作流程如图 5-18 所示：

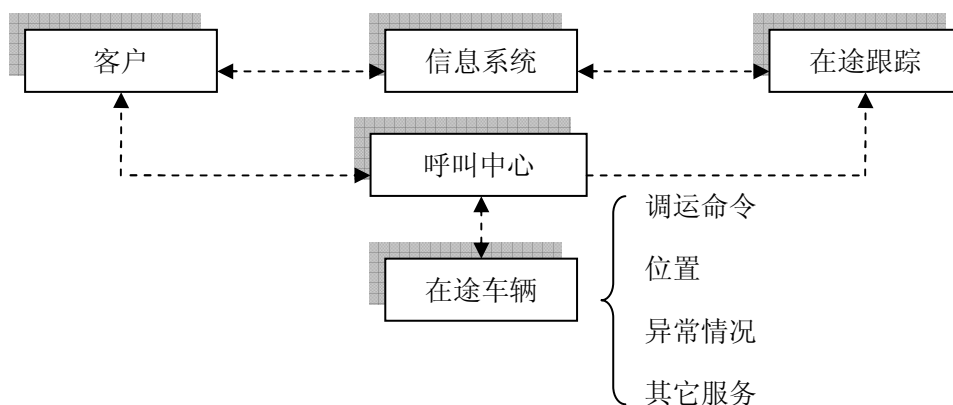


图 5-18 呼叫中心流程图

客户通过电话向呼叫中心了解货物的情况，如果司机及时反馈在途情况，呼叫中心应该立刻转告给客户。如果司机没有及时反馈在途情况，呼叫中心的跟踪员应及时联系司机，通过对司机的在途跟踪了解货物的途中情况然后再反馈给客户。

5.3.6 送达服务交割与结算流程的标准化

(一) 送达服务与交割

1. 送达服务与交割流程，其标准化流程如图 5-19 所示：

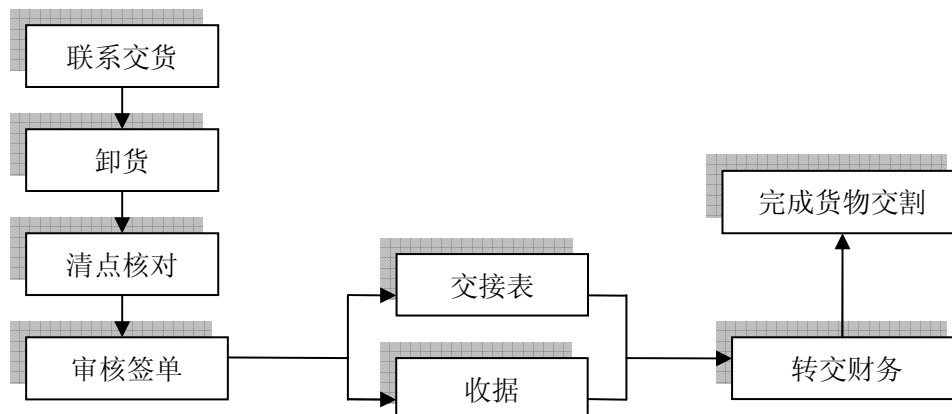


图 5-19 送达服务与交割图

2. 流程说明：

当货物送达要货地点后，送货人员应协助收获单位将货品卸下车，放到指定位置，并与收获人员一起清点货物，做好送货完成确认工作（送货坚守回单）。如果有退货、调货的要求，则应随车带回退调商品，并完成有关单证手续。

(二) 费用结算

配送部门的车辆按指定的计划完成配送工作后，即可通知财务部门进行费用结算。其标准化流程如图 5-20 所示：

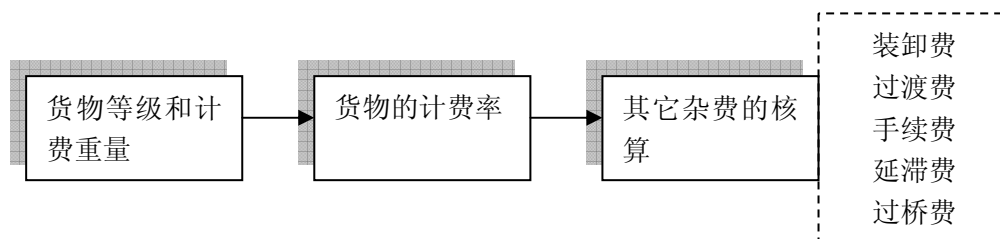


图 5-20 计费费用图

1. 计费公式有以下两种：

$$\text{运费} = (\text{货物计费重量} \times \text{计费里程} \times \text{运价率}) + (\text{货物计费重量} \times \text{计费里程} \times \text{运价率} \times \text{加成率}) \quad ①$$

$$\text{运费} = (\text{货物计费重量} \times \text{运价率}) + (\text{货物计费重量} \times \text{运价率} \times \text{加成率}) \quad ②$$

①和②区别是，①是以“吨公里”计费，②是以“吨”计费。另外，若是车辆无法计算里程或者车辆速度难以测定，计费办法是按时间计算。

2. 特种货物计费。如表 5-4 所示：

表 5-4 特种货物计费表（按特定运价来计算）

特种货物	运价方式
托运易碎、超长（货物长度 ≥ 7 米），烈性货物	质量计费
超重（每件货物重量 ≥ 250 公斤），及轻泡货物	整车计费
同一托运人托运的双程运输货物	运价率的 85% 计费
超重货物	运价加成 30% 计费
烈性货物	运价加成 110% 计费
过境公路运输	全程包干计费或者合同条款规定计费
特大型货物	协商议价
同一托运人以去程和返程运送所装货物包装	运价的 50% 计费

3. 收款方法

运杂费的收款办法主要有：

- （1）预收费用方法；
- （2）采用现金结算方法；
- （3）托收结算方法；
- （4）预交转帐支票方法。

5.4 安得公司运输问题解决方案

5.4.1 对“铜川 J 客户盈利率低”问题提出的解决方案

（一）盈利率问题的分析

铜川 J 客户自 2005 年 9 月运作至今，业务规模还可以，但盈利能力却不尽人意，毛利率比较低，加上税金和其他费用，公司实则处于亏损状态。出现该问题的原因如下：

1. 西安路线和其他路线是公司亏损的主要源头，其他地区的盈利能力也比较低。这些线路的发运特点是货量小而散，不擅于掌握车辆资源，从而导致公司的发运成本过高。
2. 调车渠道较为单一，基本依赖双生通宏货运部，对其他线路的发运渠道掌控不力，无法形成价格优势，导致发运成本偏高。
3. 亏损原因一个方面是发运货量小，基本都是零担运输；另一方面是网点对零担业务的成本和资源的掌控不够。

（二）用标准化流程提出解决方案

1. 方案简介

根据运输决策流程图的基本理念是运用信息技术对订单进行分析处理，针对不同的结果做出预测，决定是否接受订单，最大程度地避免了亏损。接受订单以后，从运货量、运输线路、调运中心等方面进行综合考虑，提高获利能力。

2. 铜川 J 客户盈利情况分析。如表 5-5 所示:

表 5-5 铜川 J 客户盈利情况分析

	应计收入	应计成本	装卸成本	毛利	毛利率
合计	2188893.55	2053642.06	41000	94251.49	4.31%

从表中可以看出,公司的盈利能力不强,如果加上税金和其他费用,公司实际处于亏损状态。从各条路线还可以看出,西安线路和其他线路是公司的亏损原因所在,下一步的改进重点应该放在西安路线和其他路线。如图 5-21 所示:

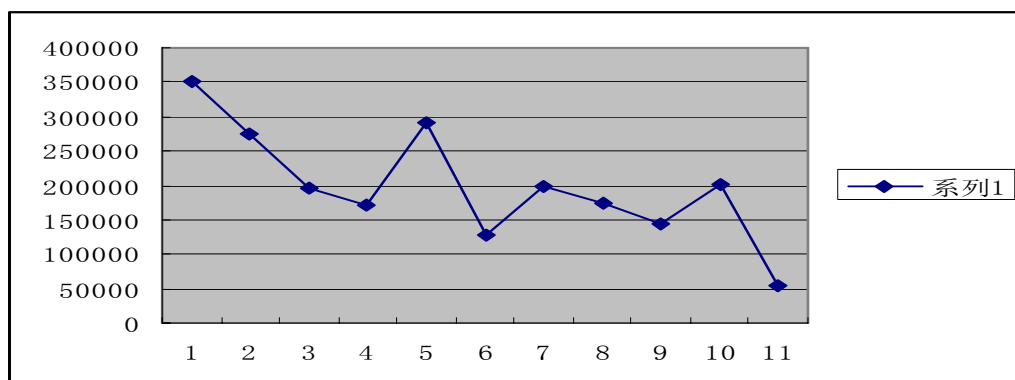


图 5-21 应计收入月度分析图

从图中可以看出,总体业绩呈现波动下滑趋势。

案例中提到,西安线有 3 辆车固定发运,成本稳定,没有下降空间。兰州线采用直发,车辆直接到厂装货。乌市线有时找不到车时到西安中转,其他线路主要是自调车或者到西安中转。调车渠道非常单一。到西安统一调车能够增强公司的管理能力,降低运行成本,贴近市场。由于兰州线和乌市线运行稳定,因此,我们主要调整的路线集中在西安路线和其他路线。

铜川 J 客户运输线路图如图 5-22 所示:



图 5-22 铜川 J 客户运输线路图

3. 运输决策流程图

运输决策流程如图 5-23 所示：

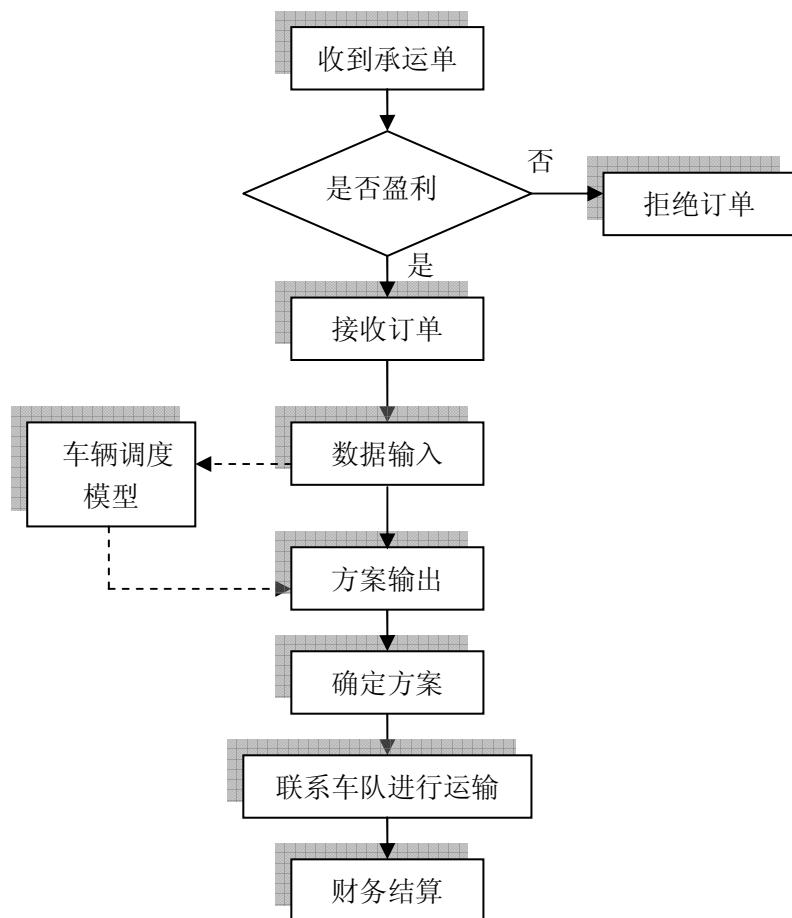


图 5-23 运输决策流程

运输决策流程图说明：

(1) 业务员根据定单进行财务分析，判断是否盈利，如果盈利则接收订单，如果没有盈利不接收订单。

(2) 业务员输入订单信息并处理数据：

根据情况在以下三种运输安排中进行选择，如图 5-24 所示：

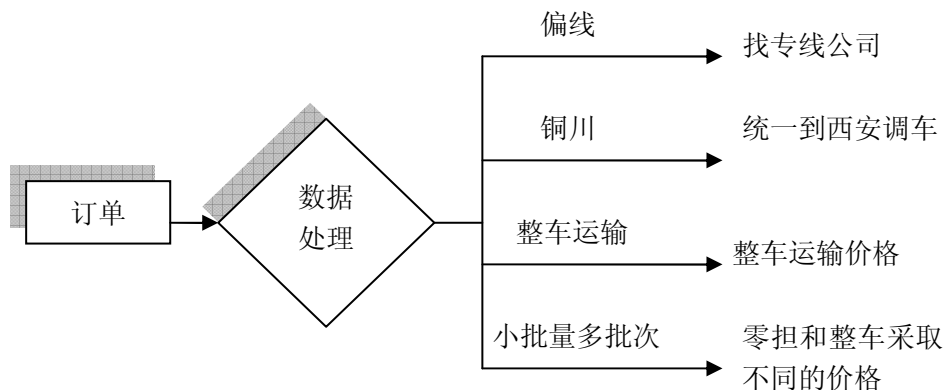


图 5-24 运输安排流程图

(3) 车辆调度过程

采用“基于配送网络和调运优先准则的物流敏捷调动优先技术”对西安路线和其他路线进行调度，车辆调度程序的示例见附录三（第 113 页）。

根据结果输出，我们可以知道车辆的空载率降到 15%到 5%之间，低于社会平均空载率 30%到 10%之间。

(4) 联系车队并进行运输

偏线路线联系专线公司，其他的线路到西安统一调车，经过车辆调度之后，安排车辆的运输线路和运输时间。并为下一步的运输计划做好准备。

4. 运输成本评估

(1) 空载率：车辆空载容积 / 车辆总容积

经过改进后的铜川 J 客户运输标准化流程设计，可以使车辆的空载率降低到 15%到 5%之间，充分利用的车辆的装载能力，减少运输成本。

(2) 配送延迟率：配送延迟车次 / 配送总车次

车辆调度过程是基于交货时间内的动态规划流程，在交货时间内把货物送达客户手中，使车辆的配送延误率得到较大的减少，提高服务的质量。

5.4.2 对“要么扭亏，要么下课”问题提出的解决方案

(一) 零担运输问题的分析

零担运输是一块常常令物流公司头疼的业务。在安得物流公司与佳得环境电器公司达成的公路运输中，从广州到河北的运输业务在 2006 年 1-3 月份亏损近 8 万元，客户对及时装货、及时到货方面也表示非常不满意。其中原因包括油价上涨、河北地区地理特殊性、客户的销售地区结构变化等。从案例得知，零担的成本比整车的高出 20%，为了集拼而压单和多次装卸；因为调车价格过高，安得公司又为寻找更低成本的承运车辆而耽误时间。因此，安得公司要减少该业务的运输成本，就要大力管理好集拼，尽量减少压单、时间延误等现象。

(二) 用标准化流程提出解决方案

1. 方案简介：

越库运输（Cross Docking）的基本理念是将收到的货物从进货车辆直接分配转移到送货车辆上。在该系统中，仓库充当库存的协调点而不是库存的储存点，物料从供应商到达仓库，然后直接转移到运输车辆上，进而尽可能快地把物料运送给用户。物料在仓库中停留的时间很短，通常不超过12个小时。这种接驳式转运方式基本上取消了仓库的储存流程，只保留合并和运输的流程，降低了仓储和运输成本。

2. 越库运输（Cross Docking）的标准运作流程如图 5-25 所示：

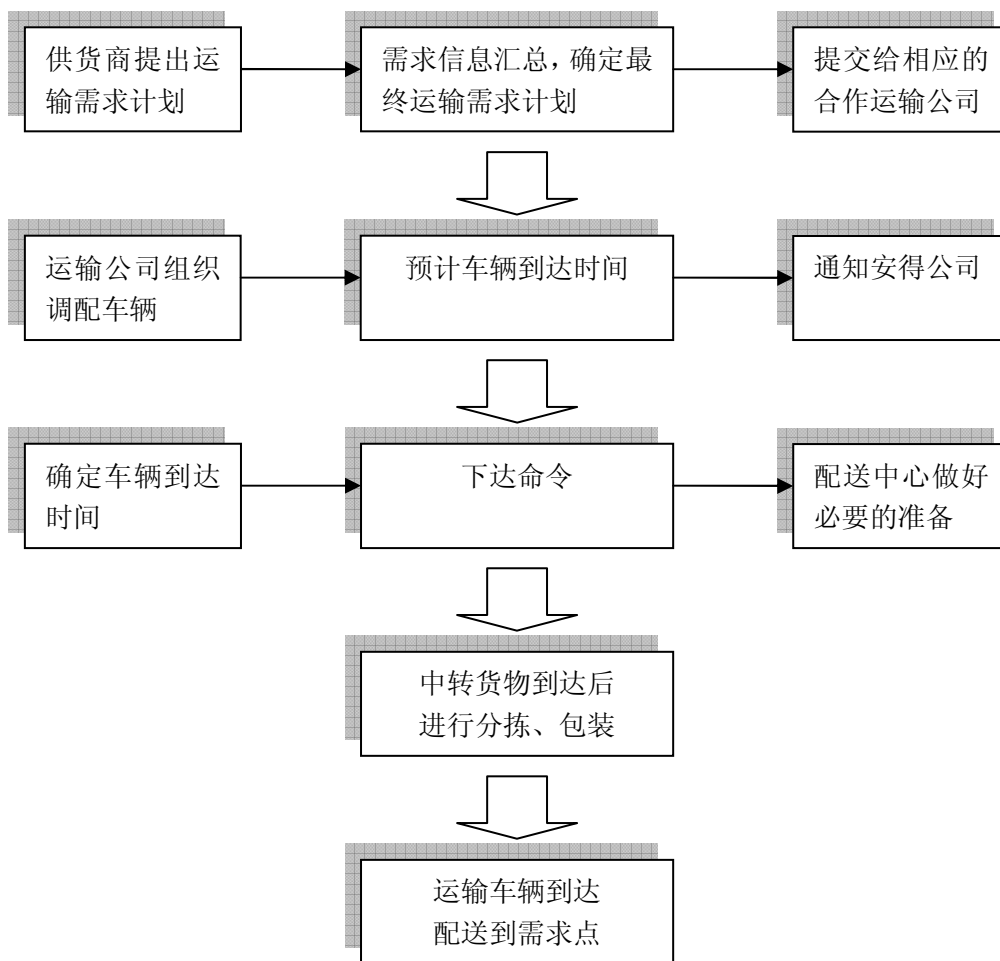


图 5-25 越库运输（Cross Docking）标准运作流程图

3. 越库运输（Cross Docking）的运作流程（结合案例中的问题）说明：

（1）供货商提出运输需求计划，安得公司的信息系统将需求信息汇总后，交给有关管理部门进行审核确定，然后将最终运输需求计划提交给相应的合作运输公司。运输方式可以采取陆运、海运等，其选择主要根据需求信息决定。

由于秦皇岛、唐山等地区的货物时间要求不高，且货运量达到集装箱运输满箱的采用海运的方式进行，安得公司将海运的需求信息汇总后，经过审核确定后，将最终运输需求计划提交给相应的合作海运公司。运作流程如图 5-26 所示：

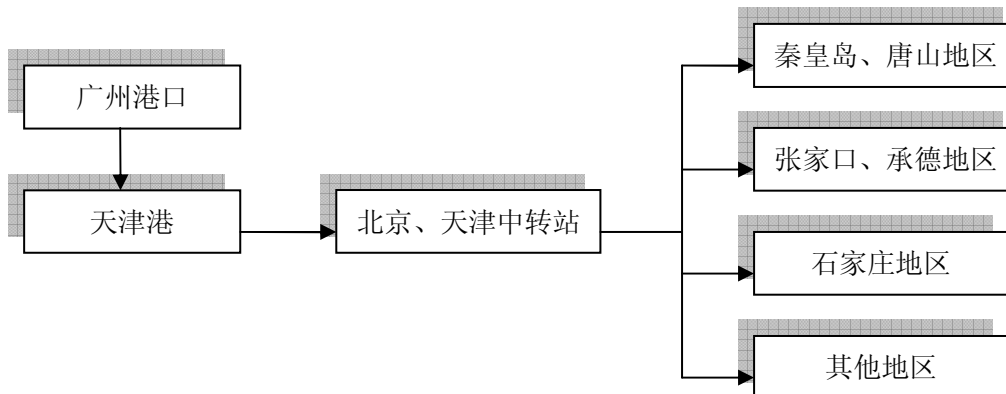


图 5-26 越库运输运作流程的实例应用

(2) 合作的运输公司收到安得公司的运输需求订单后, 根据需求计划, 做好相应的准备, 较快地组织调配车辆, 同时预计货车到达的时间, 并向安得公司的信息系统发送通知。

越库运输 (Cross Docking) 的运作关键在于三个快速: 快速收货、快速分拣和快速装车。由于中转需要停留一定的时间, 又要保证货品及时送达接收地, 因此, 要做好这项工作, 也可以利用**车辆调度模型**的程序来进行 (见附录一和附录二), 以此来提高安得公司车辆的效率、准确性和及时性。

(3) 安得公司的信息系统接收到通知后, 运用 JIT 的思想, 根据货车到达的时间来下达命令让配送中心做好必要的准备工作, 避免不必要的货物滞留和车辆等待。

要想客户配合, 必须用“利益共享”的原则。因为订单分类以后, 所需的服务被分级, 对资源的利用也会提高, 所以在成本上也是有所节省的。这样利益必须由双方共享, 才能调动客户的积极性。这种策略上的利润共享从战术上讲是细分客户需求, 利润共享从战略上来讲是建立双方的伙伴关系, 案例中的谭经理就打算走海运时给经销商每个柜 500-1000 元的补偿, 来获取经销商的接受和配合。

(4) 到货后根据客户的运输需求, 进行分拣、包装等一系列工作。

(5) 车辆到达拣货区, 装载货物后, 配送到需求点。通常要求在越库运输 (Cross Docking) 场地的停留时间不超过 24 小时, 而最快时可以在 1 小时内通过。

越库运输 (Cross Docking) 的利润点: 这种“接驳式转运”是“枢纽概念”的一种具体体现, 是指把尽可能多的货物集中到一个点, 然后进行分拣, 交叉理货, 然后进行下一次的运输。这样运作的主要目的不是减少每一次货物的运输里程, 而是每一次运输的规模的最大化, 使单位运价下降。

运用越库运输 (Cross Docking) 后的成本分析, 如表 5-6 所示:

表 5-6 地区货物流量分布 (单位: 立方米)

月份	秦皇岛、唐山地区	张家口、承德地区	石家庄地区	其他地区
1 月	341.60	619.15	747.25	427
2 月	281.16	499.84	484.22	296.78
3 月	393.58	608.26	429.36	357.8
小计	1016.34	1727.25	1660.83	1081.58

这样, 通过越库运输 (Cross Docking) 的标准化运作流程, 再配合海运运输, 总的运输成本如下:

- ① 月份订单小于 20 立方米/票占 35%, 假设从广州到天津港的运输量为整箱运输, 占总运量的 65%, 因为可以通过天津和北京中转点保存一定的库存, 起到缓冲的作用, 从而使公路运输量减少。

海运运输费用有: 每个柜取 65 方 ($(60+70)/2=65$)

$$5486 \times 65\% \div 65 = 55$$

$$55 \times (3500 + 1500) = 275000 \text{ 元}$$

- ② 假设公路运输车辆的装载量 20 方, 广州发往天津和北京的运输成本是 90-95 元/立方米, 这里取 92.5 元/立方米

公路运输费用:

$$5486 \times 35\% \times 95 = 182409.5 \text{ 元}$$

- ③ 中转费用和装卸搬运费用: $5486 \times 35\% \times (28 + 6) = 65283.4 \text{ 元}$

- ④ 暂存仓储费用可以通过公路运输的灵活性保持在适量的范围内, 不占用仓储

费。

⑤ 总的运输费用为： $275000 + 182409.5 + 65283.4 = 522692.9$ 元

盈利为： $548600 - 522692.9 = 25907.1$ （不含管理费用和税金）

公司原先亏损 78755 元到改进方案后的盈利 25907.1 元，说明了运用越库运输（Cross Docking）的标准化运作流程对运输过程进行控制保证，将能使公司成功地扭亏为盈。

因此，越库运输（Cross Docking）标准化运作流程的实践能够提升物流管理水平，提升供应链的整体运作效率与竞争力。

5.4.3 对“呼叫中心”问题提出的解决方案

（一）呼叫中心问题分析

呼叫中心成立以来，对安得公司的整个运输环节的跟踪监控取得了明显的效果，在客户和承运商之间搭起了有效的沟通桥梁，演绎了“全程掌控”的新理念。但是由于呼叫中心是安得公司的新生物，所以运作流程不是很规范，也没有制订相关的运作流程标准，在运行过程中存在一些问题，特别是虚假信息的存在和各个网点管理不规范导致服务水平难以监控，监控不到位，面对突发的事件司机缺少处理事件的流程和标准，导致司机既做运动员，又做裁判员的局面。所以对于在途运输常遇到的情况制订一个标准化的处理流程和处理标准，将提高呼叫中心的效率和客户服务水平，同时也降低了运输过程中的异常成本。

（二）用标准化流程提出解决方案

1. 方案简介

方案从一些在途运输常遇到的情况出发，制订一个标准化的处理流程和标准，如司机在途情况反馈流程、监控司机虚假信息的流程、跟踪员与司机失去联系的处理流程、异常事故处理流程、客户查货流程等有效地监控和规范司机在途情况的管理。运用这些标准流程将提高呼叫中心的效率和客户服务水平，同时也降低了运输过程中的异常成本。

2. 呼叫中心运作的具体流程

（1）司机在途情况反馈流程，其流程如图 5-27 所示：

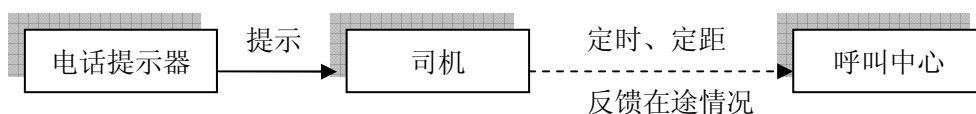


图 5-27 在途情况反馈流程图

流程说明：在车上安装电话提示器，定时或定距地提醒司机向“呼叫中心”打电话反馈在途情况，以便呼叫中心根据实际情况做出适当的调度。改变原来由呼叫中心的处理员根据需要打电话给司机的流程，规定司机定时或定距地给“呼叫中心”打电话，主动地反馈在途情况，有效地规避和减少了因跟踪不及时造成的运营风险。

（2）监控司机虚假信息的处理流程，其流程如图 5-28 所示：

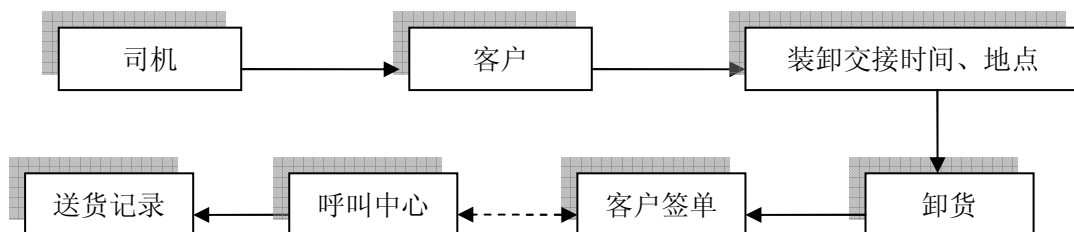


图 5-28 监控司机虚假信息处理流程

流程说明：司机在运送货物时联系客户，约好装卸交接的时间和地点，然后配合客户卸货。卸货完毕后核对单据，然后由客户签单验收。呼叫中心及时向客户了解司机送货服务的质量或者直接由客户反馈服务信息给呼叫中心，然后呼叫中心人员根据情况登记司机的送货记录作为考核资料。

(3) 跟踪员与司机失去联系的处理流程，其流程如图 5-29 所示：

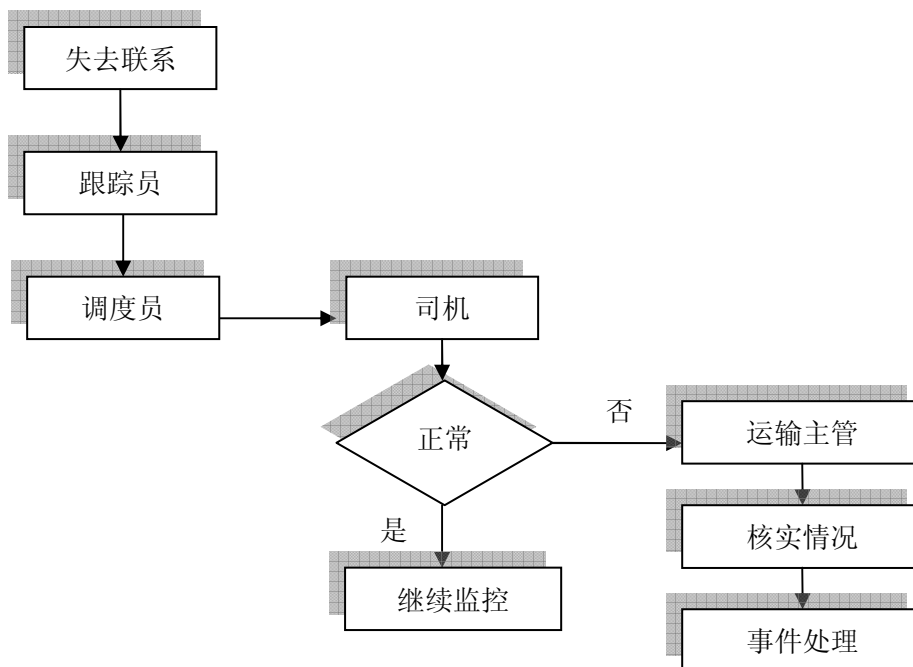


图 5-29 跟踪员与司机失去联系的处理流程图

流程说明：监控过程中出现跟踪员联系不上司机的情况，跟踪员应该及时跟调度员联系，让调度员通过各种办法尽量与司机取得联系，由司机跟呼叫中心汇报在途情况并核实在途情况顺利，跟踪员继续监控。若联系不上司机或者得知在途中出现特殊情况应该向运输主管报告然后核实情况并进行处理。

(4) 异常事故处理流程，其流程如图 5-30 所示：

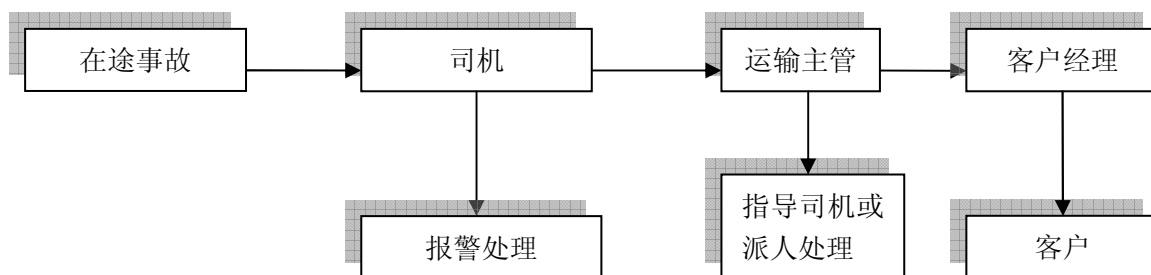


图 5-30 在途事故处理流程图

流程说明：呼叫中心跟踪员在接到司机在途的事故报告时，要求司机采取措施保护货物并及时报警，同时向运输主管报告，主管向客户经理汇报，经理马上跟客户取得联系并向客户反映事故和处理情况。运输主管在安排人员处理事故同时要有关车辆及货物资料备案。

(5) 客户查货流程，其流程如图 5-31 所示：

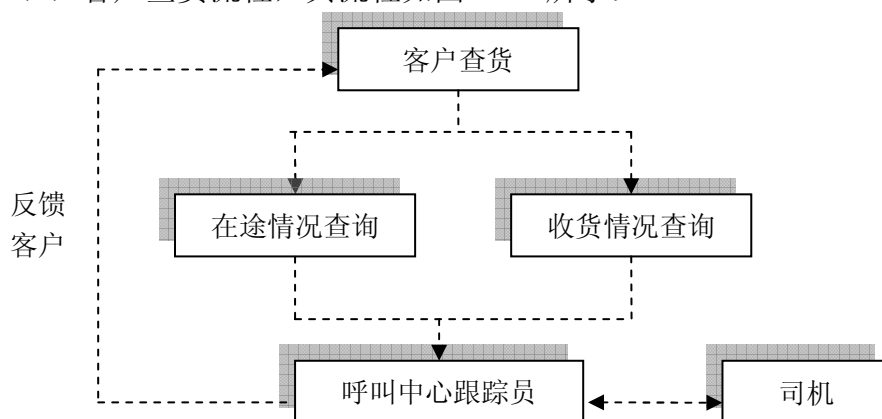


图 5-31 客户查货流程图

流程说明：客户查货过程，可以分为在途情况查询、收货情况查询。呼叫中心跟踪员负责在途情况和收货情况查询。负责接待查货的跟踪人员要根据系统的程序解答客户的需求，并做好记录作为备案。

综合上述流程，可以加强对在途车辆的监控，为在途监控提供一个标准化操作流程。从而以真实、及时的信息来保证客户服务质量。

6 物流标准化运作流程方案的实施效果评价

每个方案在实施前都要分析和评价该方案可以取得什么样的效果,然后针对实施后的结果与预期目标做出比较,再适当地对特殊情况做出事后控制。同时,由于在实施标准化运作流程时,可能会改变原来的运作流程和人们习惯的操作方式,从而引发实施过程中的一些问题,如操作人员的阻力,流程实施前后的冲击,成本的控制等,导致方案实施后没能达到预期的效果,因此,物流标准化运作流程设计方案在实施前进行效果分析与评价非常重要。在实施前,通过收集方案实体的相关数据,并利用这些数据进行仿真试验(即事前控制),就能有效地降低方案的风险,避免方案的实施过程中所造成的损失。

6.1 评价的目标

方案评价的目标要和企业的总目标一致,根据安得公司的目标“一流的物流服务品牌,专业的物流服务平台”,我们把物流标准化运作流程方案的实施效果的评价目标概括为:提高客户服务水平,提高作业效率,降低物流成本。

6.2 评价的指标

物流运作流程标准化的目的是提高物流企业服务水平,通过对各个运作流程的操作制订出一系列的标准,不仅形成了物流作业和服务质量的保障体系,而且还减少了作业各个环节的转换衔接时间、流通停滞时间和各功能模块操作时间,有效地提高了作业效率。通过运作流程的某些指标的评价可以及时地发现企业运行过程中存在的问题,及时做出改进,提高客户服务水平。

方案的评价指标将从仓储、配送、运输三个方面进行,根据案例的具体情况来分析实施物流标准化运作流程的效果。

(一) 仓储标准化运作流程评价指标

1. 进出货作业评价指标

人均劳动生产率: $\text{单位时间装卸货物数量} / \text{单位时间装卸人员数}$

进货工作率: $\text{单位时间进货量} / \text{单位时间进货人员数}$

出货工作率: $\text{单位时间出货量} / \text{单位时间出货人员数}$

2. 存货作业评价指标

储区面积率: $\text{储区面积} / \text{物流中心建筑面积}$

储位容积使用率: $\text{存货总体积} / \text{储区总容积}$

存货管理费用率: $\text{库存管理费用} / \text{平均库存金额}$

货物残损率: $\text{货物残损数} / \text{货物总数}$

3. 盘点作业评价指标

盘点数量误差率: $\text{盘点误差量} / \text{盘点总量}$

盘点品种误差率: $\text{盘点误差品种数} / \text{实施盘点品种数}$

在仓储管理的运作过程中,通过人均劳动生产率、进货工作率、出货工作率这三个指标可以衡量仓储管理的平均作业效率的高低,同时也可以评价一个标准

化流程方案是否可以提高仓储作业效率。货物残损率和盘点误差率可以衡量作业质量的高低。我们的设计方案在解决案例中“仓储设计”问题时，通过流程的标准化，使得人均劳动生产率、进货工作率、出货工作率都有所上升，这证明我们的方案是可以提高仓储作业的效率的。

（二）配送标准化运作流程评价指标

1. 订单处理作业评价指标

订单处理及时率：及时处理次数 / 总处理次数

订单延迟率：延迟交货订单数 / 订单数量

2. 成本控制作业评价指标

零担比率：零担货物量 / 总货物量

配送成本比率：（自有车配送成本+外车配送成本） / 物流总费用

3. 客户服务评价指标

客户投诉率：客户投诉次数 / 订单的数量

反馈及时率：及时反馈次数 / 总反馈次数

准时送货率：送货准时次数 / 总送货次数

缺货率：接单缺货数 / 总接单次数

配送出错率：出错次数 / 总作业次数

在配送管理的运作过程中，通过订单处理及时率和延误率两个指标可以衡量订单作业效率和质量高低。客户投诉率、准时送货率、缺货率、配送出错率和反馈及时率都可以衡量客户服务的水平。我们的设计方案在解决案例中“共同配送”、“增值服务”问题时，通过流程的标准化可以提高了订单处理的及时率、减少了零担比率和保证准时送货率。

（三）运输标准化运作流程评价指标

1. 运输作业评价指标

运输周转率：运输总车次 / （自有车数量+外车数量）×出车天数

事故率：事故发生次数 / 总服务次数

异常成本率：异常成本 / 物流总费用

延误率：延误发生次数 / 总服务次数

货物损坏率：货物损坏数量 / 货物总量

2. 车辆利用能力评价指标

空车率：空车走行距离 / 运输总距离

空载率：车辆空载容积 / 车辆总容积

在运输管理的运作过程中，通过运输周转率、事故率、空载率和异常成本率指标可以衡量成本控制能力的高低。延误率和货物损坏率可以评价客户服务水平的高低。根据这些指标，可以看到我们的设计方案在解决案例中“铜川J客户盈利率低”、“要么扭亏、要么下课”问题时，通过对流程的标准化来减少空载率和异常成本率，提高运输成本的控制能力和降低货物损坏率和减少货物的延误率，提高了客户服务水平。

6.3 评价的方法

物流标准化运作流程方案的实施效果评价可以采用事前事后关键指标比较法对方案实施前后的若干关键指标数据进行分析比较，或者用层次分析法、模糊

综合评价方法等对方案实施的物流绩效进行系统综合评价。我们认为在企业实际应用时事前事后关键指标比较法易掌握、易操作，但的缺点是方案实施需要时间资金人工等投入，万一方案不妥而实施失败会给企业带来难以预见的损失。所以我们提出借助物流仿真软件运用计算机仿真方法对物流运作流程进行模拟，跟据模拟结果进行方案实施效果分析和评价。

计算机仿真方法是指利用电子计算机对一个拟议中的或现有的系统的结构和行为进行动态模仿，并对模仿结果进行分析，以改进现有系统，设计出更优良的系统。

采用计算机仿真处理和分析实际系统有以下的优点：

1. 计算机仿真在有限的计算机空间和时间上再现真实世界的变化，从而避免了在真实系统中试验所造成的损失。
2. 借助于系统的仿真，可以对系统模型的变化过程进行反复的观察，并通过改变系统的环境、数据后，分析由此造成的影响，以便确定系统运行的最佳方案。
3. 对于一些复杂的大系统如果采用对真实系统检验的方法往往会带来很大的风险，计算机仿真这种“事前检验”的方法将能很好地解决这类问题。

我们用计算机仿真的软件模拟了仓库出库流程，以此来说明实行标准化运作流程的效果。

在仿真软件 Flexsim 环境下，原来的仿真设计图如图 6-1，改进后的仿真设计图如图 6-2 所示，仿真时间： $T=1001.875$ 。

原来的流程如图 6-1 所示：

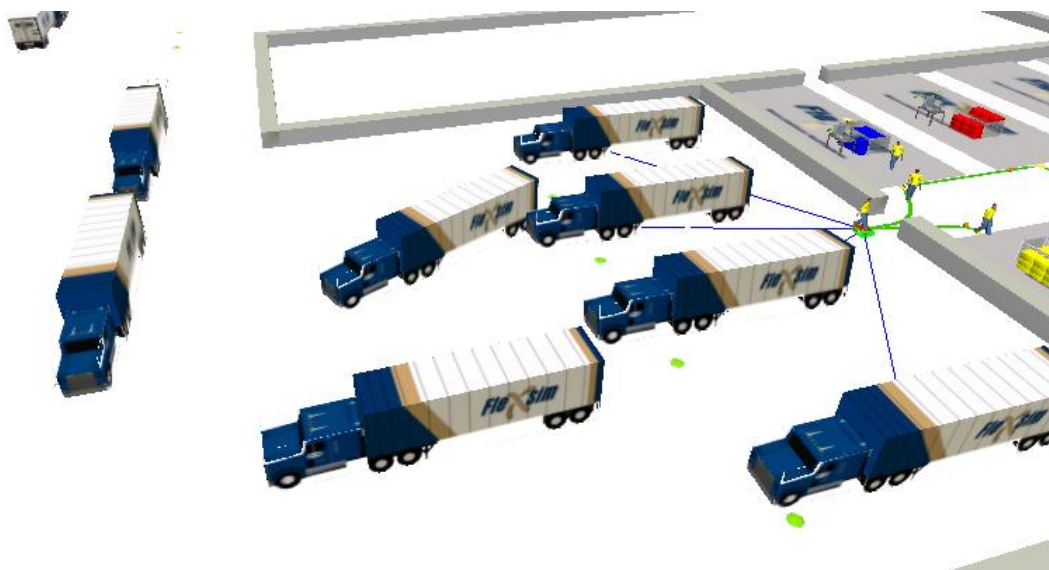


图 6-1 原出库流程

图 6-1 是按照安得公司的案例，利用物流仿真软件 Flexsim 模拟了在设立拣货区前进行装车作业的截图，图中各车辆正处于一个出库的状态，可是由于仓储出库流程设置不合理，而导致仓储运作效率低下，车辆需要交叉等待。根据仿真标准表报输出，可以得到以下数据，如表 6-1 所示：

表 6-1 原设计图输出数据

装卸点	完成装货车辆	平均等待时间
装卸点 1	52	14.91573
装卸点 2	49	14.133044
装卸点 3	72	12.200515
装卸点 4	72	11.287728
总计	245	52.537017

从输出的数据我们可以看出，在仿真时间 $T=1001.875$ 内，完成装货车辆的总数是 245 辆，出现的等到时间 52.537017。

下面我们看看改进后的流程。

改进后的流程如图 6-2 所示：

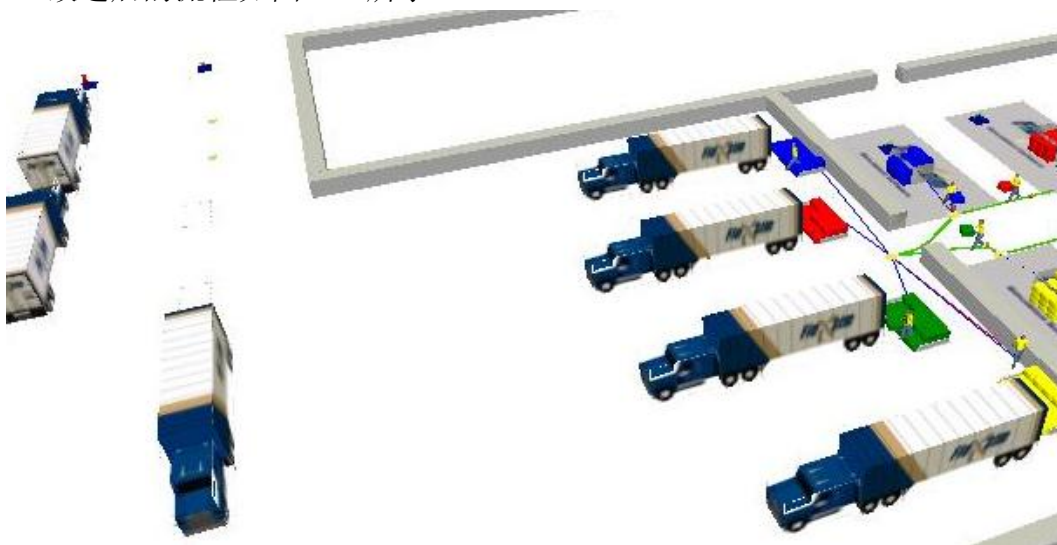


图 6-2 改进后的出库流程

图 6-1 是设立拣货区后的模拟装车作业截图。经过对搬运工审核机制的改进并同时在仓库里设立一个拣货区，由于有了拣货区，各部车辆只需要到相应的区域进行装车便能较快地完成出库流程，这样，分拣的工作便不再是出库流程的瓶颈。大部分的分拣作业在车辆到达之前已经得到完成且堆放在拣货区内，而车辆不再受到分拣作业的影响而导致等待时间过长。改进后的数据输出如表 6-2 所示：

表 6-2 改进后设计图输出数据

装卸点	完成装货车辆	平均等待时间
装卸点 1	72	8.522072
装卸点 2	75	7.831561
装卸点 3	78	8.582363
装卸点 4	73	8.922698
总计	298	33.858694

从输出的数据我们可以看出，在仿真时间 $T=1001.875$ 内，完成装货车辆的总数是 298 现的等到时间 33.858694。

从原流程仿真图的数据和改进后仿真图的数据可以看出，经过标准化流程设计，使仓库作业在相同的时间内完成较多的车辆装货，并使等待时间得到减少，说明了方案的可行性。

7 结束语

“万事开头难”。在实施标准化运作流程的时候，企业肯定会遇到许多障碍。我们为企业实施标准化运作流程提出以下建议：

1. 将新标准中有关内容和规定编入有关的操作手册，做好宣传普及和教育工作，使有关人员都能了解、熟悉、掌握这个标准。
2. 将标准化运作流程可视化，使各流程容易执行，避免出错、遗漏、颠倒等现象，比如在各个工作窗口中，用醒目的标志标明下一个工作流程是什么。
3. 制订严格的、规范的操作流程，提高作业的效率。
4. 密切关注物流技术和设备的发展，及时运用高效率的工具。
5. 提高员工的素质，认同企业的文化。
6. 营造良好的人才培养和发展的环境。

企业所面临的问题和形势并非一成不变的，标准化的制订也不是一蹴而就的，今天制订的作业标准也许明年就不适合企业面临的新情况。所以在执行标准化运作流程时要采用 PDCA 循环不断地改进，推进企业的发展和提高企业的经营效率。按照“确定方案——执行方案——评估结果并改进方案——进入下一个改进循环”的步骤循环以跟踪改进方案的实施情况，适应环境的变化，推进企业的发展和提高企业的经营效率。

公司在实施标准化方案过程中可以运用 ABC 的业务流程分析法对操作流程进行分析。ABC 的业务分析主要包括：是否存在物流作业时间上的不合理（即作业过于集中在某一段时间）；物流作业人员是否存在等待时间过长的现象；有没有物流作业与信息系统脱节的情况。如果发现上述情况的出现，立即对物流作业流程进行改善，合理配置人员，最终实现作业的平准化、效率性和建立良好的作业环境。

在此次的方案设计中，由于能力有限，加上时间仓促，我们设计出来的方案可能有许多不足的地方，但是，我们通过努力，对安得公司的运营进行了实地的参观考察，利用网络和图书馆搜索、整理和分析大量相关的资料，尽量设计出一套不仅能够适用于安得公司，也可以为其他同类企业提供参考的标准化运作流程。希望这份方案，对解决安得公司现存的问题有一定的参考价值，对安得公司的运作流程有一定的借鉴意义。

感谢教育部高校物流类教育指导委员会、全国大学生物流设计大赛组织委员会、安得物流有限公司为我们提供了这次学习钻研的机会，通过这次比赛，我们提高了实践与合作能力，对企业、对物流标准化运作流程都有了更深刻的理解，相信物流设计大赛会一次又一次为物流学习者提供一个提升的教育平台，并且一次比一次办得更好！

参考文献

- [1] 徐翔《现代企业业务流程标准化操作手册》北京：企业管理出版社 2006. 5.
- [2] 李亨《物流业实战 2000 版 ISO9001 标准》北京：中国计量出版社 2003. 9.
- [3] 丁力《第三方物流企业运作管理》长沙：湖南科学技术出版社 2003. 5.
- [4] (加) 诺曼·洛夫冰《企业流程设计指南》北京：机械工业出版社 2005. 4.
- [5] 张建伟《物流运输业务管理模板与岗位操作流程》北京：中国经济出版社 2005. 1.
- [6] 王霄涵《物流仓储业务管理模板与岗位操作流程》北京：中国经济出版社 2005. 1.
- [7] 李广泰《仓储与物料管控》深圳：海天出版社 2005. 1.
- [8] 潘林岭《现场管理实战》广州：广东经济出版社 2001. 9.
- [9] 张远昌《仓储管理与库存控制》北京：中国纺织出版社 2004. 8.
- [10] 韩岗《如何进行仓储物料管理》北京：北京大学出版社 2004. 4.
- [11] 王焰《配送中心规划与管理》长沙：湖南人民出版社 2006. 3.
- [12] 吕军伟《物流配送业务管理模版与岗位操作流程》北京：中国经济出版社 2005. 1.
- [13] 孙宏岭《高效率配送中心的设计与经营》北京：中国物资出版社 2002. 4.
- [14] 许胜余《物流配送中心管理》成都：四川人民出版社 2002. 9.
- [15] 姚城《物流配送中心规划与运作管理》广州：广东经济出版社 2004. 1.
- [16] 刘志强 丁鹏 盛焕烨《物流配送系统设计》北京：清华大学出版社 2004. 2.
- [17] 余福茂 肖亮《动态物流联盟系统规划技术研究》北京：航空工业出版社 2005. 12.
- [18] 张晓萍《物流系统仿真原理与应用》北京：中国物资出版社 2005. 5.
- [19] 孙宏岭 戚世钧《现代物流活动绩效分析》北京：中国物资出版社 2001. 12.
- [20] 霍云福 刘洪斌 孙云早《现代物流解决方案设计原理》北京：人民交通出版社 2005. 5.
- [21] 牛鱼龙《物流企业操作指南》深圳：海天出版社 2004. 7.

附录

附录一 物流敏捷调动优先技术的假设和计算

基于配送网络和调运优先准则的物流敏捷调动优先技术

1. 支持物流敏捷调动优化的基础信息描述

(1) 配送中心货物库存信息

配送中心（或调运中心，Depot）的货物库存信息由以下三元数组 A 表示：

$$A=(i, g_i, q_i)$$

其中， i 表示货物编码， $i=1,2,3,\dots$ ，简记为 $i \in I$ ； g_i 表示货物 i 的规格（单位重量、积等），并假定所有货物换算为同一规格； q_i 货物 i 的库存数量。

(2) 配送点需求信息

配送点在 t 时刻的实时需求信息由需求矩阵 $C(t)$ 来表示

$$C(t)=(c_{ij}(t)) \quad i \in I, \quad j \in J$$

其中， j 表示配送点编码， $j=1, 2, 3, \dots$ ，简记为 $j \in J$ ； $c_{ij}(t)$ 为 t 时刻配送点 j 对货物 i 的需求量， t 即交货期。

(3) 配送网络（可行运输路径）信息

配送中心至配送点 j 的配送网络的信息由以下三元数组 $R(j)$ 表示

$$R(j)=(r_j, V(r_j), D(r_j))$$

其中， r_j 表示配送中心至配送点 j 的可行运输路径编码， $r_j=1, 2, 3, \dots$ ，简记为 $r_j \in R_j$ ；

$V(r_j)$ 表示路径 r_j 上全部配送点的集合，且 $V(r_j)=\{j \mid \text{若配送点 } j \text{ 位于可行运输路径 } r_j \text{ 之上, } j \in J\}$ ；

$D(r_j)$ 表示路径 r_j 上的配送点到配送中心的距离集，且 $D(r_j)=\{d(v|r_j) \mid v \in V(r_j)\}$ ，

其中， $d(v|r_j)$ 表示在路径 r_j 上配送点 v 到配送中心的距离，显然它不一定使配送点 v 到配送中心的最短距离。至于任意配送点 v 到配送中心的最短距离则为

$$D(v)=\min\{d(v|r_v) \mid r_v \in R_v\}$$

基于以上描述的方法，配送网络所包含的信息，因可行路径搜索深度和广度而异。这里提出的基于配送网络和调运优先准则的物流敏捷调运技术的基本前提，就是假定配送中心至其服务区域内所有配送点的可行配送路径已经优化。

(4) 配送中心运输资源（车辆）信息

配送中心的运输车辆信息有以下三元数组 E 表示

$$E=(p, u_p, w_p)$$

其中， p 表示车辆编码， $p=1,2,3,\dots$ ，简记为 $p \in P$ ； u_p 表示车辆容量，并假定其已经换算为与 g_i 同一量纲； w_p 表示车辆运行状态， $w_p \in \{0, 1\}$ ，且 $w_p=1$ 表示车辆处于可调运状态；反之，则表示车辆处于正在运输状态。

2. 物流敏捷调运优化技术的工作逻辑与思想

(1) 调运优先准则

“调运优先准则”是在物流调运中根据交货期迟早、距离远近、运输量大小以及客户重要性程度等信息而对当前全部客户的物流调运优先次序进行的分类、排序。根据调运优先准则设定配送点的当前调运优先级，进一步可以构成物流调运的递阶控制机制。调运优先准则以实现物流调运敏捷化为策略层面的首要准则，并按调运策略的层次顺序

设置：

第一级调运优先准则：交货需求早的配送地要优先安排；

第二级调运优先准则：在交货期相同的配送地中,距配送中心远的优先安排；

第三级调运优先准则：针对确定的配送地,配送车辆吨位大的优先安排；

第四级调运优先准则：针对确定车辆,装载物品质量大的优先安排地方。

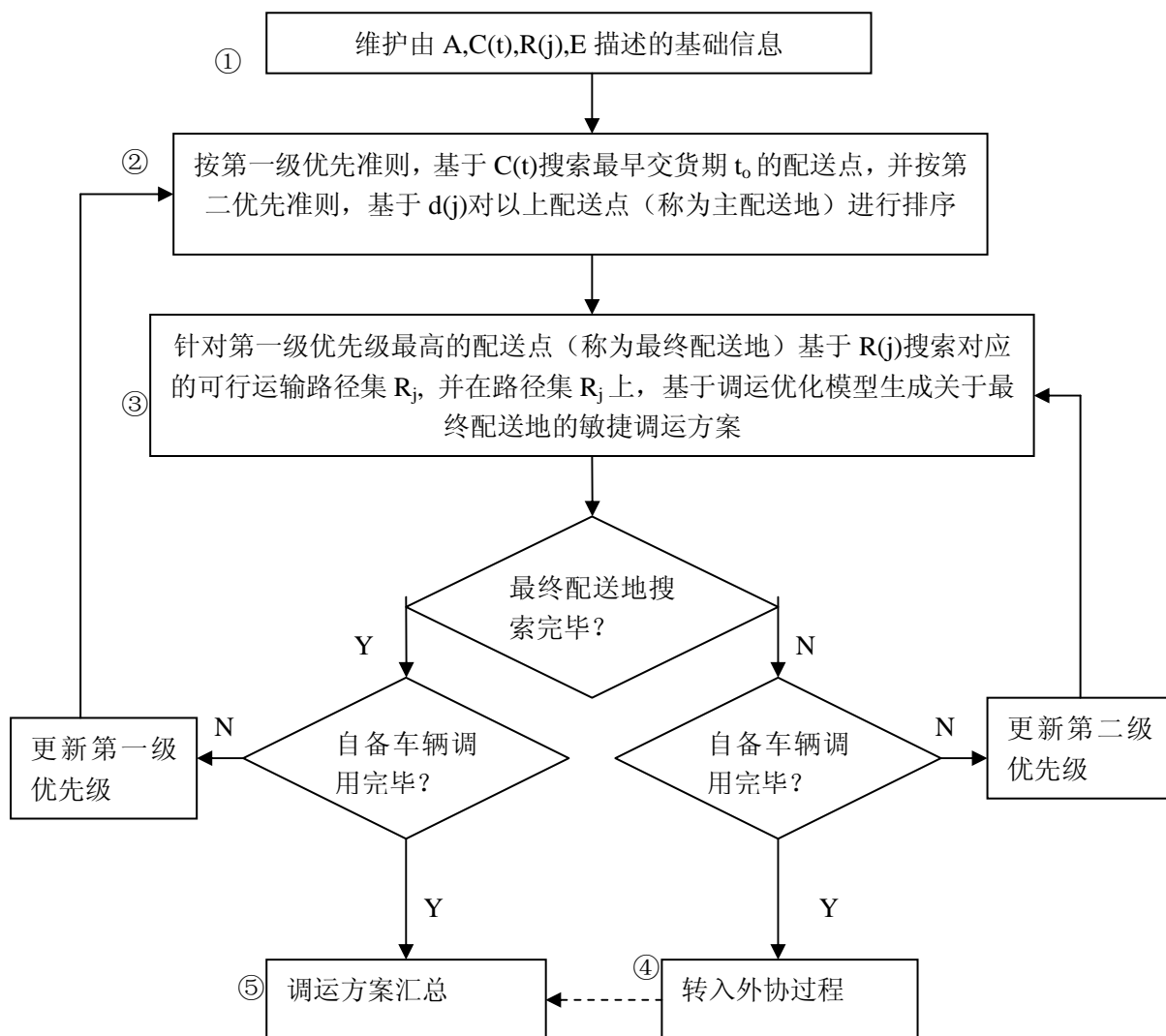


图 I 物流敏捷调动优先技术工作逻辑

从图 I 可以看出,“最终配送地”是在最早交货期 t_o 有有交货需求且距离配送中心最远的配送点,而且是动态产生的。而实现物流敏捷调动的核心在于第三个工作环节:基于调运优化模型生成关于“最终配送地”的敏捷调运优化方案。

注意到针对确定的“最终配送地” j^* 所对应的可行运输路径集构成一个以配送中心为起点,认 j^* 为终点的子配送网络。因此,整个逻辑整个配送网络的调运优化实际是转换成对一系列处于第一优先的子配送网络按第二优先准则的逐次调运优化。在调运方案生成后,对基本信息进行更新。

(2) 针对确定“最终配送地”调运敏捷化步骤（即③）：

第一步 基于 E 生成当前可调用车辆集 P^* , $P^* = \{p \mid w_p = 1, p \in P\}$ 。

第二步 针对由配送中心到最终配送地 j^* 的每一条路径 r ($r \in R_{j^*}$)

搜索该路径上在时间 t_0 有交货需求的配送点，并按第二优先准则对其排序，记为 $j^{(h)}$ （其中， $j^{(1)}$ 记为最终配送地 j^* ）。基于路径 r 上配送点到配送中心的距离集 $D(r)$ 生成与 $j^{(h)}$ 相对应的距离集 $\{d(j^{(h)} \mid r)\}$ ，记为 $\{d^{(h)}\}$ ，同时基于 $C(t)$ 和当前库存量生成配送地 $j^{(h)}$ 的需求矩阵 $C^{(h)}(t_0) = [c_{ih}^*]$ 。

第三步 针对各路径 r ，在满足 $j^{(h)}$ 在 t_0 交货的前提下，以“空载率”极小化为目标，建立调运优化模型，进行车辆配置优化和货物配载敏捷化。

第四步 在针对各路径 r 的调运优化模型求解后，将对应“空载率”最小的路径选出作为配送中心向“最终配送地” j^* 配送货物的“优选路径”，对应该路径的优化配送方案则为关于 j^* 的敏捷调运方案，同时更新有关基础信息。

第五步 转到工作逻辑下一个工作环节，直到符合终于原则。

3. 针对确定路径调运优化模型的建立及近似求解方法。

(1) 针对确定路径 r 的调运优化模型的建立。

①决策变量的确定

设关于车辆的选择变量为 $t_{hl} \in \{0, 1\}$, $h \in v^*(r)$, $l \in P^*$ 其中， $t_{hl} = 1$ 表示第 l

号车被派往第 h 号主配送地 $j^{(h)}$ ，反之不派。

记 x_{hlk} 为第 l 号车派往第 $j^{(h)}$ 号主配送地时，第 k 种货物的装载量。

②调运优化模型

由于调运优化的目标是“空载率”极小化，注意到路径 r 的空载率 $\eta^{(r)}$ 为

$$\eta^{(r)} = 1 - \left(\sum_{h \in v^*(r)} \sum_{l \in P^*} \sum_{k \in I} d^{(h)} t_{hl} x_{hlk} \right) / \left(\sum_{h \in v^*(r)} \sum_{l \in P^*} d^{(h)} t_{hl} u_l \right) \quad (1)$$

而第 $j^{(h)}$ 号主配送地所需求货物总量 (GG_h) 的计算公式为：

$$GG_h = \sum_{i \in I} c_{ih}^* g_i, \quad h \in v^*(r) \quad (2)$$

这里， $v^*(r)$ 为路径 r 上在时间 t_0 有交货需求的配送点集。于是，在库存和装载限制下，

满足货运需求 GG_h 的针对确定路径 (r) 的调运优化模型可由如下的非线性混合型规划问题③~⑧描述，记为 NP

$$\text{NP: } \max \left(\sum_{h \in v^*(r)} \sum_{l \in P^*} \sum_{k \in I} d^{(h)} t_{hl} x_{hlk} \right) / \left(\sum_{h \in v^*(r)} \sum_{l \in P^*} d^{(h)} t_{hl} u_l \right) \quad (3)$$

$$\text{St. } \sum_{l \in P^*} \sum_{k \in I} d^{(h)} t_{hl} x_{hlk} = GG_h, \quad h \in v^*(r) \quad (4)$$

$$\sum_{h \in v^*(r)} \sum_{l \in P^*} t_{hl} x_{hlk} \leq q_k^*, \quad k \in I \quad (5)$$

$$\sum_{h \in v^*(r)} \sum_{k \in I} t_{hl} x_{hkl} \leq u_l^*, l \in p^* \quad (6)$$

$$t_{hl} \in \{0,1\}, h \in v^*(r), l \in p^* \quad (7)$$

$$x_{hkl} \geq 0, h \in v^*(r), l \in p^*, k \in I \quad (8)$$

其④为需求约束条件，⑤为库存约束条件，其中 q_k^* 为货物 k 的当前库存量；⑥为资源约束条件；其中 u_l^* 为车辆 l 的剩余装载量。

(2) 基于确定路径调运优化模型近似求解的基本过程：

我们对调运人格化模型 NP 近似求解的基本思想是按确定路径 r 上主配送地（即在 t_0 有交货需求的配送点）从远到近的顺序（即第二级优先准则），通过数据驱动将 NP 逐次分解为针对配送点 $j^{(h)}$ 的调运优化模型 NP_h 后，再对其递阶求解。该求解过程如下：

基于确定路径调运优化模型的递阶求解过程：

过程 1：赋初值

令 $h=1, Fl_{h-1}=0, P_0 = \Phi$

过程 2：数据驱动过程

Step1：基于需求矩阵 $[C_{ih}^*]$ 和当前库存量 $q_k^* (k \in I)$ ，生成针对 $j^{(h)}$ 的满足库存限制的需求集和缺货信息并修正 GG_h

Step2：基于 P^* 和上一级优化后生成的“车辆优选集”（已调用的车辆） P^{h-1} ，生成针对 $j^{(h)}$ 的“当前可调用车辆集” P_h^* 。

过程 3：模型生成过程：

基 NP 和过程 2 的数据驱动生成针对确定主配送地 $j^{(h)}$ 的调运优化模型⑨～⑬，记为 NP_h 。

$$NP_h : \max \left(\sum_{l \in p_h^*} \sum_{k \in I} d^{(h)} t_{hl} X_{hkl} \right) / \left(\sum_{l \in p_h^*} d^{(h)} t_{hl} U_l \right) \quad (9)$$

$$\text{St.} \quad \sum_{l \in p_h^*} \sum_{k \in I} d^{(h)} t_{hl} X_{hkl} = GG_h \quad (10)$$

$$\sum_{l \in p_h^*} \sum_{k \in I} d^{(h)} t_{hl} X_{hkl} \leq U_1^* \quad (11)$$

$$t_{hl} \in \{0, 1\}, l \in P^* \quad (12)$$

$$x_{hkl} \geq 0, l \in P^*, k \in I \quad (13)$$

过程 4： NP_h 求解过程

Step1：基于 NP_h 求得针对主配送地 $j^{(h)}$ 的“车辆优选集” P_h 及相应的最佳装载量，并计算优先级最低的车辆的剩余装载量 Fl_h 。

Step2：令 $h=h+1$ ，若 $j^{(h)} \in V^*_{(r)}$ ，则转过程 2；反之，则结束对 (NP) 的递阶求解过程。

可以看出，经过“过程 2”之 Step1 的数据处理所生成的 NP_h 将自然删除关于库存约束的约束条件式⑤，再经过“过程 2”之 Step2 的数据处理使得 NP_h 的规模进一步有效缩小。尽管如此， NP_h 仍然是一个非线性混合型规划，关于 NP_h 求解过程（“过程 4”之 Step1）下面讨论。

4、针对确定主配送地的调运优化模型 NP_h 的求解：

(1) 为了更有效地实现空载率极小化的目标，根据 NP_h 的模型的具体含义，其求解采用用

基于模型驱动和数据驱动相结合的优化过程。如下：

过程 1：数据驱动过程

Step1: 记 NP_{h-1} 求解后优先级最低车辆(不妨记该车车号为 $P^{(h-1)}$)的“剩余装载量”为 Fl_{h-1} 。然后在 Fl_{h-1} 的限制下，基于需求矩阵 $[C_{ih}^*]$ 并根据第四级级准则对该车进行配载，随即修正需求矩阵 $[C_{ih}^*]$ ，记修正后的需求矩阵为 $[C_{ih}']$ 。同时，计算针对主配送地 $j^{(h)}$ 的剩余装载需求量 FF_h

$$FF_h = GG_n - Fl_{h-1}$$

Step2: 对“可调用车辆集” P_h^* 按第三级优先准则排序，记排序后的“可调用车辆集”为 P_h' ，记 P_h' 中各车辆相应装载容量为 U_l' ($U_1' > U_{l+1}'$, $l \in P_h'$)

过程 2：模型生成过程

基于过程 1 的数据驱动，将 NP_h 转化成为“剩余装载量”极小化为目标，以 P_h' 和 FF_h 为配载约束且仅以关于车辆选择变量 t_{hl} 为决策变量的 0-1 规划模型(14)~(16)。

$$(0-1)_h: \min \sum_{l \in P_h'} (t_{hl} U_l' - FF_h) \quad (14)$$

$$\text{St.} \quad \sum_{l \in P_h'} t_{hl} U_l' \geq FF_h \quad (15)$$

$$t_{hl} \in \{0, 1\}, l \in P_h' \quad (16)$$

过程 3：求解过程

Step1: 对 $(0-1)_h$ 求解，并基于“最优解”生成的“优选车辆集” P_h ，计算优先级最低车辆的“剩余装载量” Fl_h

Step2: 基于需求矩阵 $[C_{ih}']$ 和满载要求，先按第三级优先准则，再按第四级优先准则，以数据驱动方式对“优选车辆集” P_h 中的车辆进行配载，生成 x_{h1k} 的“最优解”。

(2) 值得说明的 n 个问题：

- ①关于 0-1 规划的求解比较成熟，后面讨论；
- ②对优先级最低车辆的剩余装载量的继续配载。

5. 基于对分搜索 $(0-1)_h$ 的求解过程：(如图 II 所示)

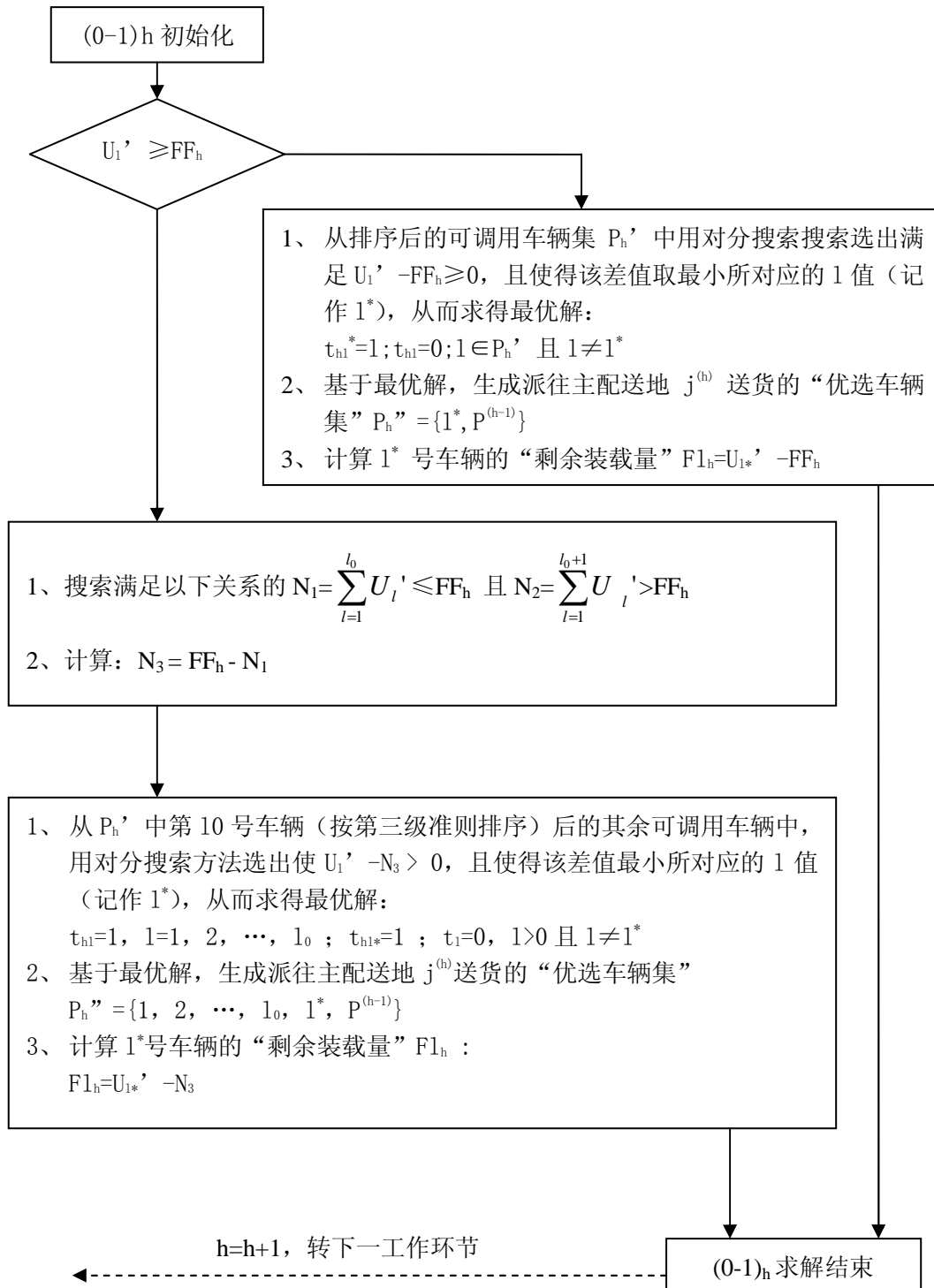


图 II 基于对分搜索 (0-1) _h 的求解过程

附录二 物流敏捷调动优先技术在 Visual C#环境下编程和操作说明

(一) 物流敏捷调动优先技术在 Visual C#环境下编写的程序源代码如下:

```
using System;
using System.Collections;
using System.Text;

namespace GreedyThr
{
    class BubbleSorter //-----冒泡排序类-----
    {
        static public void Sort(object[] sortArray, Op2 gtMethod)
        {
            for (int i = 0; i < sortArray.Length; i++)
            {
                for (int j = i + 1; j < sortArray.Length; j++)
                {
                    if (gtMethod(sortArray[j], sortArray[i]))
                    {
                        object temp = sortArray[i];
                        sortArray[i] = sortArray[j];
                        sortArray[j] = temp;
                    }
                }
            }
        }
    }

    delegate bool Op2(object l, object r);
    //定义一个比较类委托

    class Marx //-----寻找最短路径类-----
    {
        private int[] distance;
        private int row;
        private ArrayList ways = new ArrayList();

        public int Get_distance(int distance_index)
        {
            return this.distance[distance_index];
        }
    }
}
```

```
public Marx(int n, params int[] d)
{
    this.row = n;
    distance = new int[row * row];
    for (int i = 0; i < row * row; i++)
    {
        this.distance[i] = d[i];
    }
    for (int i = 0; i < this.row; i++) //有 row 个点，则从中心到各点的路
有 row-1 条
    {
        ArrayList w = new ArrayList();
        int j = 0;
        w.Add(j);
        ways.Add(w);
    }
}
//-----
public void Find_way() //基于 Dijkstra 算法求各点到配送中心的最短路径
{
    ArrayList S = new ArrayList(1);
    ArrayList Sr = new ArrayList(1);
    int[] Indexof_distance = new int[this.row];

    for (int i = 0; i < row; i++)
    {
        Indexof_distance[i] = i;
    }

    S.Add(Indexof_distance[0]);

    for (int i = 0; i < this.row; i++)
    {
        Sr.Add(Indexof_distance[i]);
    }
    Sr.RemoveAt(0);
    int[] D = new int[this.row]; //存放中心点到每个点的距离

    //-----以上已经初始化了，S 和 Sr(里边放的都是点的编号)-----
    int Count = this.row - 1;
    while (Count > 0)
    {
        //假定中心点的编号是 0 的 Dijkstra 算法求路径
    }
}
```

```
for (int i = 0; i < row; i++)
    D[i] = this.distance[i];

int min_num = (int)Sr[0]; //距中心点的最小距离点编号

foreach (int s in Sr)
{
    if (D[s] < D[min_num]) min_num = s;
}

//以上可以排序优化
S.Add(min_num);
Sr.Remove(min_num);
//-----把最新包含进来的点也加到路径中-----
((ArrayList)ways[min_num]).Add(min_num);

foreach (int element in Sr)
{
    int position = element * (this.row) + min_num;
    bool exchange = false; //有交换标志

    if (D[element] < D[min_num] + this.distance[position])
        D[element] = D[min_num] + this.distance[position];
    else
    {
        D[element] = this.distance[position] + D[min_num];
        exchange = true;
    }
    //修改距离矩阵
    this.distance[element] = D[element];
    position = element * this.row;
    this.distance[position] = D[element];

    //修改路径-----
    if (exchange == true)
    {
        ((ArrayList)ways[element]).Clear();
        foreach (int point in (ArrayList)ways[min_num])
            ((ArrayList)ways[element]).Add(point);
    }
}

--Count;
}
```

```
//-----  
public void Display()  
{  
    //-----中心到各点的最短路径-----  
    Console.WriteLine("中心到各点的最短路径如下: \n\n");  
    int sum_d_index = 0;  
    foreach (ArrayList mother in ways)  
    {  
        foreach (int child in mother)  
        {  
            Console.Write("V{0} -- ", child + 1);  
            Console.WriteLine("    路径长 {0}", distance[sum_d_index++]);  
        }  
    }  
    public void Outway(out ArrayList a)           //输出路径, 用于使网络得到路  
    {  
        a = this.ways;  
    }  
    public void Outdistance(out int[] d)          //输出距离集  
    {  
        d = this.distance;  
    }  
}  
class Center    //配送中心  
{  
    private ArrayList storage = new ArrayList();    //库存量集  
    private int typeNum;        //货物种类  
  
    public Center(int[] s) //构造函数, 生成库存量集  
    {  
        this.typeNum = s.Length;  
        for (int i = 0; i < this.typeNum; i++)  
        {  
            this.storage.Add(s[i]);  
        }  
    }  
  
    public void dec(int q, int i)    //减去输出量  
    {  
        this.storage[i] = (int)this.storage[i] - q;  
    }  
}
```

```
public bool empty(int i)           //第 i 种货物库存量是否为空
{
    return (int)this.storage[i] == 0;
}
public void displayS()             //输出各种货物库存量
{
    int i = 0;
    foreach (int s in this.storage)
    {
        Console.WriteLine("第 {0} 种货物现存 {1} ", ++i, s);
    }
}

}

class A_point           //配送点
{
    private int index;           //点编号
    private DateTime[] t;        //时间集
    private int goodNum;         //货物种类
    private int timeNum;         //时间点个数
    private int[] require;       //需求量集, 下标与货物对应
    private int distance;        //与配送中心的距离
    private bool status;         //有需求 true, 无需求 false

    //时间与需求量有一一对应关系,
    //第一个时间点(交货日期)对各种货物的需求量

    public int GetIndex()
    {
        return this.index;
    }
    public DateTime Get_first_time()
    {
        return this.t[0];
    }
    public int GetRequire()       //获取该点第一个交货日期对各种货物的总需求量
    {
        int i = 0, sum = 0;
        while (i < this.goodNum)
        {
            sum += this.require[i];
            i = i + 1;
        }
    }
}
```

```
        return sum;
    }

    public void Set_TandR()    //配载完成后对需求矩阵的修改
    {
        if (this.timeNum <= 1)
        {
            this.status = true; //若原来只有一个交货日期，配载完成后直接设置
该点为完成状态

            int[] empty = new int[0];
            this.require = empty;
            DateTime[] emptyt = new DateTime[0];
            this.t = emptyt;
            return;
        }
        DateTime[] time_buff = new DateTime[this.timeNum - 1];
        int i = 1, j;
        for (j = 0; j < this.timeNum - 1; j++)
            time_buff[j] = this.t[i++];
        --this.timeNum;
        this.t = time_buff;    //以上几行是把 this.t 的第一个元素去掉
        int[] require_buff = new int[this.require.Length - this.goodNum];
        i = this.goodNum;
        for (j = 0; j < this.require.Length - this.goodNum; j++)
        {
            require_buff[j] = this.require[i];
            i++;
        }
        this.require = require_buff; //以上几行是对需求量集的修改，去掉了第
一个时间点需求量
    }

    public A_point(Marx m, int ind, DateTime[] ts, int[] r) //输入交货日期和
相应货物的需求量
    {
        this.index = ind;
        this.t = ts;
        this.require = r;
        this.timeNum = ts.Length;
        this.goodNum = r.Length / this.timeNum;
        this.distance = m.Get_distance(this.index);
        this.status = false;
    }
```



```
//设置对需求量进行减少的方法
public void deSet(DateTime ta, int reduce, int goodIndex)
{
    //ta 是时间点, goodIndex 是货物号, reduce 是减少量
    int i;
    for (i = 0; i < this.t.Length; i++)
    {
        if (this.t[i] == ta)
            this.require[i + goodIndex * this.t.Length] -= reduce;
    }
}

//配送完毕标志, 这里假设车辆很多, 跑一次一定能够配送完毕
public bool completed()
{
    return (this.status == true);
}

public void displayInfo()
{
    Console.WriteLine("\n 配送点 V{0} : 与配送中心的距离是 {1}, 需求矩阵
如下", this.index + 1, this.distance);
    Console.Write("      ");
    foreach (DateTime ta in this.t)
    {
        Console.Write("{0}  ", ta.ToLongDateString());
    }
    Console.WriteLine();

    int i, j;
    for (i = 0; i < this.goodNum; i++)
    {
        Console.Write("货物{0}  ", i + 1);
        for (j = i; j < this.require.Length; j += this.goodNum)
            Console.Write(this.require[j] + "      ");
        Console.WriteLine();
    }
    Console.WriteLine();
}

public static bool compare(object l, object r)
{
    A_point lm = (A_point)l;
    A_point rm = (A_point)r;
    return (lm.t[0] < rm.t[0]);
}
```

```
}
public static bool compare2(object l, object r)
{
    A_point lm = (A_point)l;
    A_point rm = (A_point)r;
    return (lm.distance > rm.distance);
}
}
class web                                //配送网络
{
    private ArrayList ways;
    private int[] points;
    private int[] distance_p;

    public void SetWays(ArrayList w)      //测试通过设置路径
    {
        this.ways = w;
    }

    public void SetDistance(int[] d)      //设置距离集合
    {
        this.distance_p = d;
    }

    public void SetPoint(int rows)        //设置点集
    {
        for (int i = 0; i < rows; i++) this.points[i] = i;
    }

    public ArrayList Get_ways()            //获得路径
    {
        return this.ways;
    }

    public void displayWeb()              //显示路径网络
    {
        int sum_d_index = 0;
        foreach (ArrayList mother in this.ways)
        {
            foreach (int child in mother)
            {
                Console.WriteLine("V{0} -- ", child + 1);
                Console.WriteLine("    路径长 {0}", distance_p[sum_d_index++]);
            }
        }
    }
}
```

```
}

class car
{
    private int index;           //车编号
    private int weight;         //车容量
    private int weight_less;     //剩余装载量，跟“空载率”挂钩
    private bool status;        //true 可用，false 不可用

    public car(int ind, int w)
    {
        this.weight = w;
        this.index = ind;
        this.status = true; //初始化车可调用
        this.weight_less = this.weight;
    }

    public int getIndex() //获取车号
    {
        return this.index;
    }

    public int getWeight() //获取车容量
    {
        return this.weight;
    }

    public int getWeight_less() //获取车剩余容量
    {
        return this.weight_less;
    }

    public void setStatus(bool sta) //设置车的工作状态
    {
        this.status = sta;
    }

    public bool getStatus() //获取车的当前状态
    {
        return this.status;
    }

    public void setWeight_least(int used) //经配载后，车剩余容量的更改
    {
```

```
        this.weight_less = this.weight - used;
    }

    public void InitW()           //送货完成后，车清空，即要还原装载量
    {
        this.weight_less = this.weight;
    }

    public static bool compare(object l, object r)
    {
        car lm = (car)l;
        car rm = (car)r;
        return (lm.weight > rm.weight);
    }

    public void display()
    {
        Console.WriteLine("{0}           {1}           {2}           {3}",
this.index, this.weight, this.weight_less, this.status);
    }
}

//-----基于车辆的二分搜索-----
class Binary_sThr           //实用型的折半查找，查找绝对值比 value 大且最接近
vaule 的值
{    //适合于从大到小排序的序列，返回值是车号
    static public int Search(car[] searchArray, int value)
    {
        int first = 0, end = searchArray.Length - 1, half;

        //数组下标从 0 开始，length 返回的是长度，所以要减 1

        if (value > searchArray[first].getWeight_less())
            return -1;           //若比最大的还大，那么无解

        if (value < searchArray[end].getWeight_less())
            return searchArray[end].getIndex();           //若比最小的还小，那么最
小那个就是所求的值

        do
        {
            half = (end + first) / 2;
            if (value == searchArray[half].getWeight_less())
                return searchArray[half].getIndex();
            else if (value < searchArray[half].getWeight_less())
            {
```

```
        if (searchArray[half + 1].getWeight_less() < value)
            return searchArray[half].getIndex();
        first = half + 1;
    }
    else
    {
        if (searchArray[half - 1].getWeight_less() > value)
            return searchArray[half - 1].getIndex();
        end = half - 1;
    }
} while (first <= end);

return -1;
}
}

//*****
class NPdeal
{
    //此类是处理类, 先获取输入数据

    private web w;           //网络
    private Center c;        //中心
    private A_point[] points; //配送点集
    private car[] cars;      //车辆集

    public NPdeal(web a_w, Center a_c, A_point[] a_points, car[] a_cars)
    {
        this.w = a_w;
        this.c = a_c;
        this.points = a_points;
        this.cars = a_cars;
    }
    //按交货期对各配送点排序
    public void Point_Sort()
    {
        Op2 Op3 = new Op2(A_point.compare);
        BubbleSorter.Sort(this.points, Op3);
    }

    public void display_Points()
    {
        foreach (A_point po in this.points)
        {
```

```
        if (po.completed() == false)
        {
            po.displayInfo();
        }
    }
}
//按容量对各车辆排序
public void Car_Sort()
{
    Op2 Op4 = new Op2(car.compare);
    BubbleSorter.Sort(this.cars, Op4);
}
public void display_Cars()
{
    Console.WriteLine("车号    容量    剩余载量    当前状态");
    foreach (car cc in this.cars)
    {
        cc.display();
    }
}
//处理
public void run1()
{
    Console.WriteLine("\n\n 原始数据(未排序):\n");
    display_Points();
    Console.WriteLine();
    display_Cars();
    Console.WriteLine();

    int PointNum = this.points.Length; //总点数
    //在下边的迭代中表示还有需求的点总数
    while (PointNum > 0)
    {
        this.Point_Sort();           // 对点排序

        //获取所有 t0 时刻有需求的点集
        ArrayList points_buff = new ArrayList();
        int i = 0;
        for (; i < this.points.Length; i++)
        {
            if (this.points[i].Get_first_time() ==
this.points[0].Get_first_time())
                points_buff.Add(this.points[i]);
        }
    }
}
```



```
//在以上点集中根据第 2 优先级排序（与中心的距离）
A_point[] start_points = new A_point[points_buff.Count];
//最终配送地 start_points, 有相同交货期，但路径没限制

int ii = 0;
foreach (A_point a in points_buff)
{
    start_points[ii++] = a;
}

Op2 Op5 = new Op2(A_point.compare2);
BubbleSorter.Sort(start_points, Op5);

int sPointNum = start_points.Length;
//当前最终配送地总数
Console.WriteLine(" 当前最终配送地是: ");
foreach (A_point a in start_points)
{
    Console.WriteLine("V{0}    ", a.GetIndex() + 1);
}
Console.WriteLine("\n");

for (int s_Index = 0; s_Index < sPointNum; s_Index++)
{
    //----- 搜索路径网络，以化为
    NPh-----
        int a_index = start_points[s_Index].GetIndex();
        DateTime          first_time          =
start_points[s_Index].Get_first_time();
        ArrayList endpoint_buff = new ArrayList();    //路径点缓存

        //要先去掉路径上的起点（即配送中心），
        //然后进行查找，找到后要重新把中心点加回到路径中
        Console.WriteLine("距离最远的配送地是 V{0} ", a_index + 1);
        //***** 这 里 修 改 中
        *****
        ((ArrayList)this.w.Get_ways()[a_index]).RemoveAt(0);
        foreach (int v in (ArrayList)this.w.Get_ways()[a_index])//
因为配送点的索引从 1
        {
            //
            开始，而配送网络路径从 0 开始，所以要减 1，这个错误找了好久
            foreach (A_point a in this.points)
            {
```

```
        if (a.GetIndex() == v)
        {
            if (a.Get_first_time() == first_time)
                endpoint_buff.Add(a);
        }
    }
}
((ArrayList) this.w.Get_ways()[a_index]).Insert(0, 0);

A_point[] end_points = new A_point[endpoint_buff.Count];
//得到同路径上有相同交货期的配送点集 end_poings
ii = 0;
foreach (A_point a in endpoint_buff)
{
    end_points[ii] = a;
    ii++;
}

Console.WriteLine("V{0} 点路径上有相同交货日期的点有\n",
a_index + 1);

foreach (A_point a in end_points)
{
    Console.Write("V{0}  ", a.GetIndex() + 1);
}
Console.WriteLine();

BubbleSorter.Sort(end_points, 0p5);
Console.WriteLine("按距离排序 : ");
foreach (A_point a in end_points)
{
    Console.Write("V{0}  ", a.GetIndex() + 1);
}
Console.WriteLine("\n");
for (int e_Index = 0; e_Index < end_points.Length; e_Index++)
{
    Console.WriteLine(" 当前主配送地为 V{0}\n",
end_points[e_Index].GetIndex() + 1);
    Console.WriteLine("\n车辆已排序: \n");
    Car_Sort();
    // display_Cars();
    int a_time_require = end_points[e_Index].GetRequire();
    int carIndex = Binary_sThr.Search(this.cars,
a_time_require);

    Console.Write("按任意键继续__");
```

```
Console.Read();
Console.Clear();

//车的二分查找已经通过!!
if (carIndex != -1)
{
    //表示一辆车已经足够
    foreach (car cr in this.cars)
    {
        if (cr.getIndex() == carIndex)
        {
            cr.setStatus(false);
            //此车已经被用
            cr.setWeight_least(a_time_require);
            //对此车进行配载, 修改剩余容量
            break;
        }
    }
}
else
{
    //一辆车还不够
    int N1 = 0;
    int indexOf_N2 = 0;
    foreach (car cr in this.cars)
    {
        if (cr.getStatus() == true)
        {
            N1 += cr.getWeight_less();
            int aless = cr.getWeight_less();
            indexOf_N2++;
            cr.setStatus(false);    //调用该车
            cr.setWeight_least(cr.getWeight());
            if (N1 > a_time_require)
            {
                N1 = N1 - aless;    //要退回一辆车
                cr.setWeight_least(0);
                //参数是已用容量, 这里相当于还原剩余容量
                cr.setStatus(true);
                indexOf_N2--;
                break;
            }
        }
    }
}

ArrayList car_buffer = new ArrayList();
//当前可用车辆缓存
```

```

foreach (car cr in this.cars)
{
    if (cr.getStatus() == true)
        car_buffer.Add(cr);
}
car[] instanceCar = new car[car_buffer.Count];
//当前可用车辆集
int cark = 0;
foreach (car cr in car_buffer)
{
    instanceCar[cark++] = cr;
}
int N3 = a_time_require - N1;
Car_Sort();
int b_carIndex = Binary_sThr.Search(instanceCar, N3);
int Fl = 0;
foreach (car c in this.cars)
{
    if (c.getIndex() == b_carIndex)
    {
        c.setStatus(false);
        Fl = c.getWeight_less() - N3;
        c.setWeight_least(N3);
        break;
    }
}
//到这里，第一个配送点的第一个交货期需求配送完毕
}
end_points[e_Index].Set_TandR();
if (end_points[e_Index].completed())
{
    //如果这个点已经配送完，那么删掉这个点。
    Console.WriteLine("配送点 V{0} 的所有需求都已满足",
end_points[e_Index].GetIndex() + 1);
    Kill_apoint(end_points[e_Index].GetIndex());
    --PointNum;
}
//修改该配送点的需求矩阵
//***** 看 看 结 果 如 何
*****
// Console.WriteLine("\n\n 经本次配载后，各数据如下: \n");
displayAll();
//end for1
}
//end for2

```

```
    }
    //算下空载率
    double raid = 0.0, x=0.0, u=0.0;
    foreach (car c in this.cars)
    {
        if (c.getStatus() == false)
        {
            x += c.getWeight();
            u += c.getWeight();
            if (c.getWeight_less() != 0)
                x = x - c.getWeight_less();
        }
        c.setStatus(true);
    }
    raid = 1 - x / u;
    Console.WriteLine("\n 空载率:  {0}\n\n", raid);
    //end while
}
Console.WriteLine("\n\n\n^^ 全部点的需求已满足, 配送完成!\n");
}

public void Kill_apoint(int index)
{
    A_point[] buffer = new A_point[this.points.Length - 1];
    if (buffer.Length == 0)
    {    //只有最后一个点了

        return;
    }
    int i = 0;
    foreach (A_point a in this.points)
    {
        if (a.GetIndex() != index)
        {
            buffer[i++] = a;
        }
    }
    this.points = buffer;
}

public void displayAll()
{    //显示最终结果
    Console.Clear();
    Console.WriteLine("经过这一轮配载后, 各数据如下:\n");
    this.w.displayWeb();
}
```

```
        Console.WriteLine("\n");
        display_Cars();
        Console.WriteLine("\n\n");
        display_Points();
        Console.WriteLine("\n");
        Console.WriteLine("按任意键进行下一轮配送__\n\n");
        Console.Read();
    }
}

//*****
class MainEntryPoint
{
    public void Default_Input()
    {
        int r;    //列数
        r = 7;
        int[] a = new int[] {0, 2, 3, 5, 99, 99, 5,
                               2, 0, 99, 2, 99, 7, 99,
                               3, 99, 0, 99, 5, 99, 99,
                               5, 2, 99, 0, 3, 5, 99,
                               99, 99, 5, 3, 0, 2, 7,
                               99, 7, 99, 5, 2, 0, 1,
                               5, 99, 99, 99, 7, 1, 0};

        Marx m = new Marx(r, a);
        m.Find_way();
        Console.WriteLine();

        Console.WriteLine("生成路径图:\n");
        //-----
        web w1 = new web();
        ArrayList webWay1 = new ArrayList();
        m.Outway(out webWay1);
        w1.SetWays(webWay1);

        int[] b = new int[r * r];
        m.Outdistance(out b);
        w1.SetDistance(b);

        w1.displayWeb();
        int[] s = new int[] { 250, 350 };
        Center c = new Center(s);
        Console.WriteLine("\n 配送中心: \n");
    }
}
```



```
c.displayS();
//-----
//-----测试配送点 -----
int index1 = 1;
DateTime[] ts1 = new DateTime[2];
ts1[0] = new DateTime(2007, 1, 5);
ts1[1] = new DateTime(2007, 2, 13);
int[] rs1 = new int[4] { 6, 3, 5, 7 };

int index2 = 2;
DateTime[] ts2 = new DateTime[2] {
    new DateTime(2007, 1, 5),
    new DateTime(2007, 3, 8)};
int[] rs2 = new int[4] { 8, 7, 1, 0 };

int index3 = 3;
DateTime[] ts3 = new DateTime[2] {
    new DateTime(2007, 1, 5),
    new DateTime(2007, 2, 6)};
int[] rs3 = new int[4] { 18, 7, 4, 3 };

int index4 = 4;
DateTime[] ts4 = new DateTime[2] {
    new DateTime(2007, 1, 6),
    new DateTime(2007, 2, 16)};
int[] rs4 = new int[4] { 7, 6, 14, 13 };

int index5 = 5;
DateTime[] ts5 = new DateTime[2] {
    new DateTime(2007, 1, 15),
    new DateTime(2007, 3, 6)};
int[] rs5 = new int[4] { 17, 2, 0, 1 };

int index6 = 6;
DateTime[] ts6 = new DateTime[2] {
    new DateTime(2007, 2, 5),
    new DateTime(2007, 4, 6)};
int[] rs6 = new int[4] { 0, 16, 2, 3 };

A_point[] points = new A_point[6] {
    new A_point(m, index1, ts1, rs1),
    new A_point(m, index2, ts2, rs2),
    new A_point(m, index3, ts3, rs3),
    new A_point(m, index4, ts4, rs4),
```

```
        new A_point(m, index5, ts5, rs5),
        new A_point(m, index6, ts6, rs6)};

//-----
//-----测试车辆 -----
car[] cars = new car[10]{
    new car(1, 10),
    new car(2, 13),
    new car(3, 18),
    new car(4, 27),
    new car(5, 28),
    new car(6, 20),
    new car(7, 10),
    new car(8, 11),
    new car(9, 17),
    new car(10, 15)};

//-----
NPdeal np = new NPdeal(w1, c, points, cars);
np.run1();
}
public void Self_Input()
{
    int r;    //列数
    Console.WriteLine("请输入点个数（含配送中心点）：");
    Int32.TryParse(Console.ReadLine(), out r);
    Console.WriteLine("各点分别为：\n");
    for (int i = 1; i <= r; i++)
        Console.WriteLine("V{0} ", i);
    Console.WriteLine("假定第一个点是配送中心");
    Console.WriteLine("\n\n 输入各点之间的距离(无通径的用相对大整数表
示)\n");

    int[] a = new int[r * r];
    //定义距离矩阵
    int da;

    for (int i = 0; i < r; i++) //----输入各点之间的距离
    {
        for (int j = i + 1; j < r; j++)
        {
            Console.WriteLine("V{0} 到 V{1} 的距离是: ", i + 1, j + 1);
            Int32.TryParse(Console.ReadLine(), out da);
            a[i * r + j] = da;
        }
    }
}
```

```
        Console.WriteLine();
    }
}
//----完善距离矩阵(距离矩阵其实可以是个上三角矩阵,
//----但为了处理方便,还是将其完整成一个对称阵)-----
for (int i = 0; i < r; i++)
{
    for (int j = 0; j < r; j++)
    {
        if (i == j)
        {
            a[i * r + j] = 0;
        }
        a[j * r + i] = a[i * r + j];
    }
}
Marx m = new Marx(r, a);    //新建一个找路径对象
m.Find_way();               //生成路径
Console.WriteLine("生成路径图:\n");

web w1 = new web();
ArrayList webWay1 = new ArrayList();
m.Outway(out webWay1);
w1.SetWays(webWay1);

int[] b = new int[r * r];
m.Outdistance(out b);
w1.SetDistance(b);
w1.displayWeb();

//-----路径网络创建完毕-----

//-----
Console.WriteLine("\n 输入配送中心货物数目:");
int types;
Int32.TryParse(Console.ReadLine(), out types);
int[] s = new int[types];
Console.WriteLine("输入各种货物库存量: ");
for (int i = 0; i < types; i++)
{
    Int32.TryParse(Console.ReadLine(), out s[i]);
}
Center c = new Center(s);
Console.WriteLine("\n 配送中心: \n");
```

```
c.displayS();
//-----配送中心数据输入完毕-----

//-----
int pNo = r - 1;
A_point[] points = new A_point[pNo];
Console.WriteLine("\n 录入配送点数据: \n");
for (int i = 0; i < pNo; i++)
{
    Console.Write("V{0} 点的交货期个数是: ", i + 2);
    int times = 0;
    Int32.TryParse(Console.ReadLine(), out times);
    DateTime[] t = new DateTime[times];
    string st;
    for (int j = 0; j < times; j++)
    {
        Console.Write("第 {0} 个交货日期是(输入格式如: 2007/1/1): ",
j + 1);

        st = Console.ReadLine();
        t[j] = DateTime.Parse(st);
    }
    int[] requires = new int[times * types];
    int r_index = 0;
    for (int js = 0; js < times; js++)
    {
        Console.WriteLine("第 {0} 个交货期对各货物的需求量为:
", js+1);

        for (int ty = 0; ty < types; ty++)
        {
            Console.WriteLine("货物 {0}: ", ty+1);
            Int32.TryParse(Console.ReadLine(),
requires[r_index++]);
        }
    }
    points[i] = new A_point(m, i + 1, t, requires);
}
//-----配送点录入完毕-----
//-----
Console.WriteLine();
Console.WriteLine("输入车辆数目: ");
int carNum = 0;
Int32.TryParse(Console.ReadLine(), out carNum);
car[] cars = new car[carNum];
Console.WriteLine("输入各车载重量: ");
```

```
for (int i = 0; i < carNum; i++)
{
    int carWeight;
    Console.WriteLine("车{0} :", i + 1);
    Int32.TryParse(Console.ReadLine(), out carWeight);
    cars[i] = new car(i + 1, carWeight);
}
//-----车辆录入完毕-----

NPdeal np = new NPdeal(wl, c, points, cars);
np.run1();
}
static void Main(string[] args)
{
    Console.WriteLine("***** 基于调运优先准则的物流敏捷调运优化系统
*****\n\n");
    MainEntryPoint a = new MainEntryPoint();

    Console.WriteLine("按 1、测试已有方案\n 按 2、自行录入方案\n 按 3、退出");
    int chose;
    while (true)
    {
        Int32.TryParse(Console.ReadLine(), out chose);
        if (chose == 1)
            a.Default_Input();
        else if (chose == 2)
            a.Self_Input();
        else break;
    }
}
}
```

（二）物流敏捷调动优先技术在 Visual C#环境下编程的操作说明

在 Visual C#环境下，通过新建一个控制台程序，将该程序的源代码复制过去即可运行调试。另外，在录入数据的操作过程中有一些地方做一下说明。

自己录入数据有些潜规则说明：

1. 各配送点的各交货日期的输入要先输入时间早的，再输入迟的。
2. 程序省去了一个步骤，即对优先级最低车辆的剩余装载量的继续配载。
3. 关于“当前可调用车辆集”的问题，车辆什么时候可用，什么时候不可用基于以下原理：
在当前配送地的路径上可能有多个点有相同的交货日期，若车 X 被点 P1 用了，车的状态就设置为 false，即该路径上与 P1 有相同交货日期的点 P2 就不能调用车 X，而当该路径上所有有相同交货日期的点都配送完成后，车 X 的状态才会变成可用(true)。例如在下图中，

V2 V4 V5 是一条路径上的点，若这 3 个点有相同的交货日期，按照优先级 2，应该先给 V5 配送，所以，已经被 V5 所配载装满的车，就不能被 V4 或 V2 利用。

4. 关于空载率的说明，基于上述第一点，本程序求出的不是“修正后的空载率”，而是每当一条路径上有相同交货日期的点配送完后，就根据车辆配载和剩余容量统计出的空载率。

如下图，V2 V4 V5 若有相同交货日期，则对该日期的该路径上的点 2，4，5 配送完成后，统计一下该路径的空载率。

5. 每次配载（即某一个点的一个交货期）完成后，程序会打印出剩余配送点的需求矩阵，（比较每次当前配送点的需求矩阵就可以看出差别），同时会打印出各车的各项参数，但程序有清屏功能，所以每次看屏幕要把滚动条拉到最上端再往下看。

6. 程序的默认方案如图 I 所示：

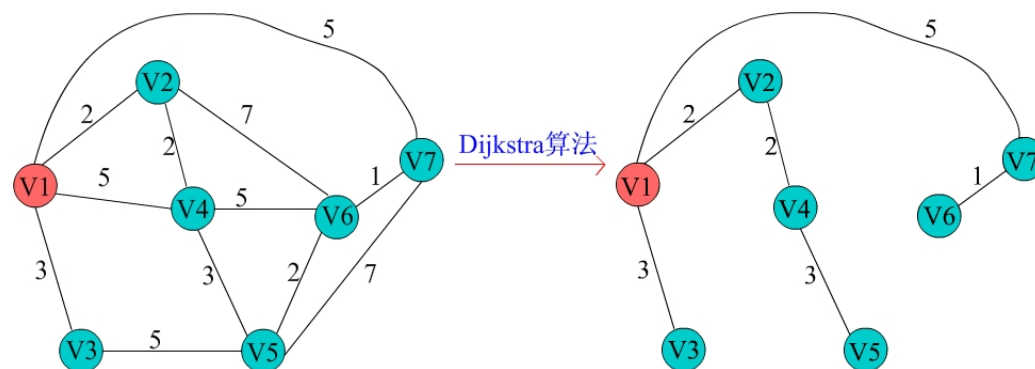


图 1 Dijkstra 算法路径图

以下为测试输出结果的截图如图 II 所示

```

经过这一轮配载后，各数据如下：

V1 -- 路径长 0
V1 -- V2 -- 路径长 2
V1 -- V3 -- 路径长 3
V1 -- V2 -- V4 -- 路径长 4
V1 -- V2 -- V4 -- V5 -- 路径长 7
V1 -- V7 -- V6 -- 路径长 6
V1 -- V7 -- 路径长 5

车号  容量  剩余载量  当前状态
5      28      0      False
4      27      0      False
6      20      0      False
3      18     18      True
9      17      5      True
10     15      0      True
2      13     13      True
8      11     11      True
7      10      3      True
1      10      5      False

按任意键进行下一轮配送__

空载率： 0.0588235294117647
    
```

图 II 测试输出结果截图

附录三 物流敏捷调动优先技术在 Visual C#环境下的应用示例

根据车辆资源调度情况，统一到西安调车，调车渠道如图 I 所示：

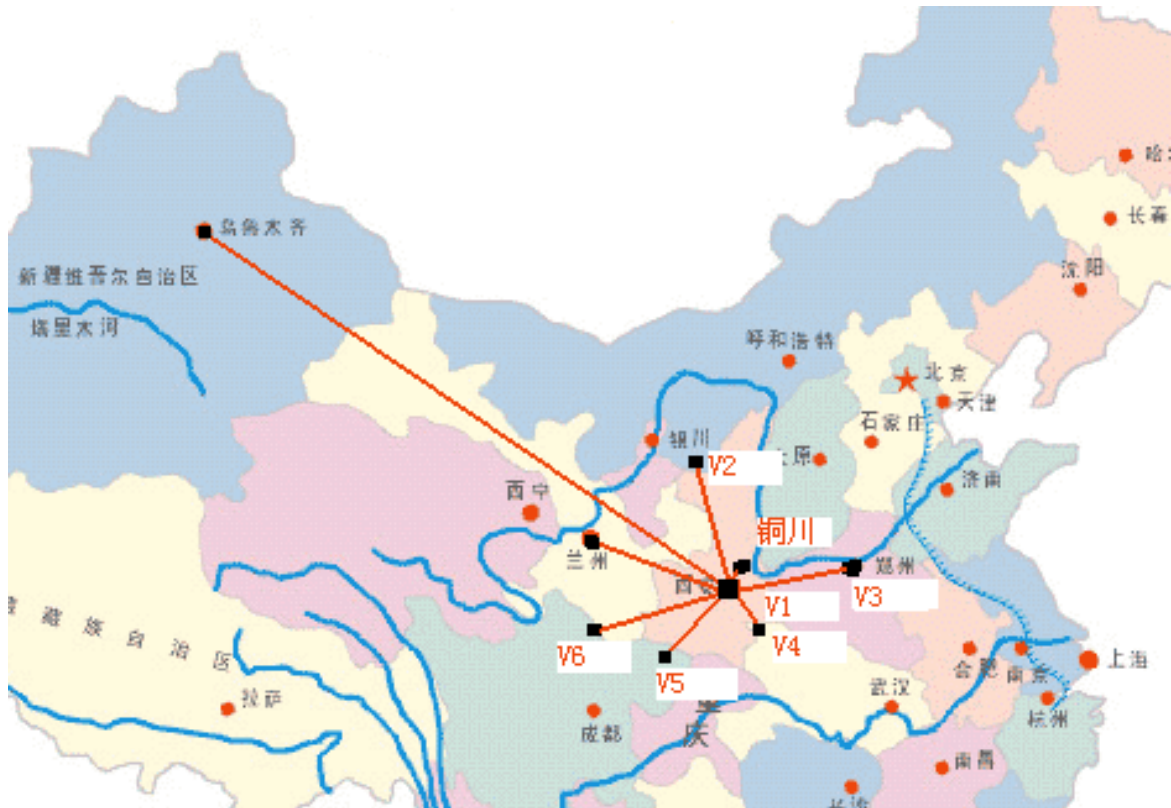


图 I 西安统一调车线路

到西安统一调车能够增强公司的管理能力，降低运行成本，贴近市场。兰州线和乌市线运行稳定，主要调整路线是西安路线和其他路线。

流程运作如下图 II 所示：

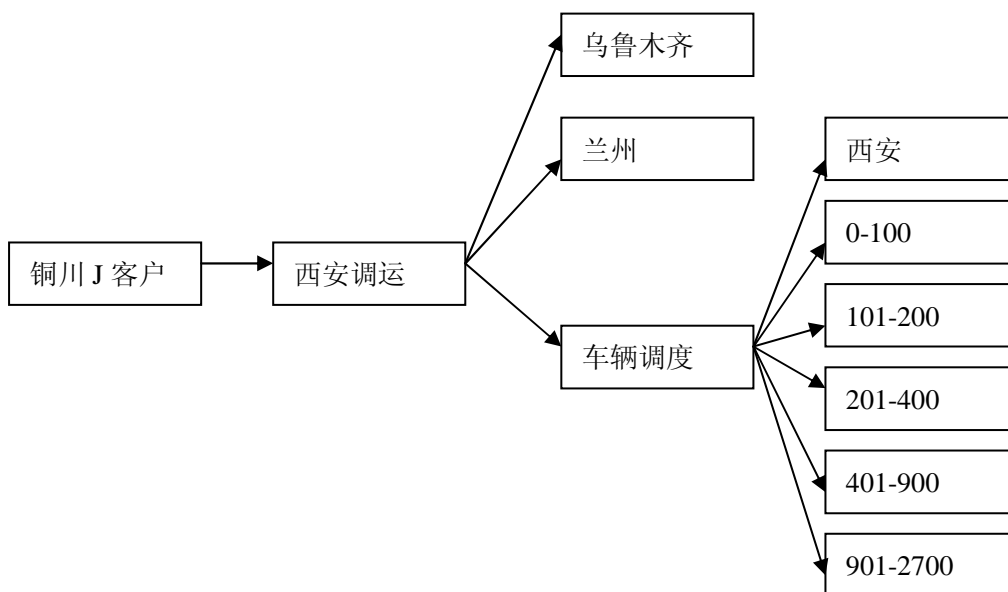


图 II 铜川 J 客户西安调车运作流程

（一）基础数据的假设与输入。

在这里我们有必要说明基础数据主要基于案例的数据，并作了适当的假设，与公司的实际情况有一定的偏差，但是不影响该模型在实际中的应用。

① 配送点数目：6 个，分别是西安、0-100、101-200、201-400、401-900、901-2700，并假设为 V_1 、 V_2 、 V_3 、 V_4 、 V_5 、 V_6 。

②各个配送点之间的距离

总地图上测得各线路之间的距离如表 I 所示：

表 I 配送点距离

线路	距离（公里）D
西安到乌市	2134
西安到兰州	534
西安到 0-100 (V_1 - V_2)	136
西安到 101-200 (V_1 - V_3)	379
西安到 201-400 (V_1 - V_4)	135
西安到 401-900 (V_1 - V_5)	310
西安到 901-2700 (V_1 - V_6)	826
0-100 到 101-200 (V_2 - V_3)	356
0-100 到 201-400 (V_2 - V_4)	739
0-100 到 401-900 (V_2 - V_5)	490
0-100 到 901-2700 (V_2 - V_6)	553
101-200 到 201-400 (V_3 - V_4)	397
101-200 到 401-900 (V_3 - V_5)	671
101-200 到 901-2700 (V_3 - V_6)	859
201-400 到 401-900 (V_4 - V_5)	278
201-400 到 901-2700 (V_4 - V_6)	567
401-900 到 901-2700 (V_5 - V_6)	340

③配送中心货物的种类数目：5(假设值，货物的种类可以根据实际情况而定)

④配送中心各个货物的库存量（假设库存量稳定），如表 II 所示：

表 II 各种货物库存量

货物的种类	库存量
Q1	40
Q2	40
Q3	60
Q4	80
Q5	40

⑤各个配送点需求货物的种类数目和每种货物的交货时间（交货时间为假设值），如表 III 所示：

表III 货物交货时间

配送点	交货时间个数	时间 t		
V ₂	2	2007/3/14	2007/3/21	
V ₃	3	2007/3/14	2007/3/21	2007/3/28
V ₄	1		2007/3/21	
V ₅	3	2007/3/14	2007/3/21	2007/3/28
V ₆	2		2007/3/21	2007/3/28

⑥在交货日期下配送点需求的各种货物的数量，如表IV所示：

表IV 各时间段货物需求量

配送点	货物的需求量 C _t		
	2007/3/14	2007/3/21	2007/3/28
V ₂	26 (Q1=5; Q2=10; Q3=11)	18 (Q1=13; Q3=5)	0
V ₃	10 (Q1=5; Q3=5)	20 (Q1=5; Q4=15)	13 (Q4=5; Q5=8)
V ₄	0	16 (Q3=10; Q4=6)	0
V ₅	13 (Q2=10; Q3=3)	12 (Q3=4; Q4=8)	20 (Q4=10; Q5=10)
V ₆	0	10 (Q4=10)	24 (Q4=14; Q5=10)

其中，26 (Q1=5; Q2=10; Q3=11) 表示配送点 V₂ 货物需求总量是 26，当中包括货物 1（即 Q₁）需求量是 5，货物 2 的需求量是 10，货物 3 的需求量是 11，货物 4 和 5 的需求量是 0。

⑦可调运的车辆数目和各车辆的载重量(车辆包括返回的可调用车辆，比如车辆 1 完成 t₁ 时间的送货，返回之后系统自动重新给车辆编码 10，成为可调用的车辆)。如表 V 所示：

表V 车辆载重量

车辆编码 p	载重量 U _p
1	15
2	15
3	15
4	20
5	20
6	20
7	20
8	20
9	15
10	15
11	15
12	15
13	10
14	15
15	10
16	10
17	10
18	10

(二) 以上基础数据输入程序之后可以得到的输出结果

1. 交货时间在 2007/3/14 的配送路线:

① V_3 ;

② V_5 ;

③ V_2 ;

配送点交货时间在 2007/3/14 的需求矩阵表, 如表VI所示:

表VI 2007/3/14 的需求矩阵表

货物	V_3	V_5	V_2
货物 1 (Q1)	5	0	5
货物 2 (Q2)	0	10	10
货物 3 (Q3)	0	3	11
货物 4 (Q4)	15	0	0
货物 5 (Q5)	0	0	0

配送点 V_3 调用车辆的编号为 13;

配送点 V_5 调用车辆的编号为 14;

配送点 V_2 调用车辆的编号为 4, 18

库存量:

空载率: 0.109

2. 交货时间在 2007/3/21 的配送路线:

① $V_6 \rightarrow V_5$;

② V_5 ;

③ V_3 ;

④ V_2 ;

⑤ V_4 ;

配送点交货时间在 2007/3/21 的需求矩阵表, 如表VII所示:

表VII 2007/3/21 的需求矩阵表

货物	V_6	V_3	V_5	V_2	V_4
货物 1 (Q1)	0	5	0	13	0
货物 2 (Q2)	0	0	0	0	0
货物 3 (Q3)	0	0	4	5	10
货物 4 (Q4)	14	15	8	0	6
货物 5 (Q5)	10	0	0	0	0

配送点 $V_6 \rightarrow V_5$ 调用车辆的编号为: 4, 12;

配送点 V_5 调用车辆的编号为: 11;

配送点 V_3 调用车辆的编号为: 5, 18;

配送点 V_2 调用车辆的编号为: 8;

配送点 V_4 调用车辆的编号为: 7;

空载率: 0.227

3. 交货时间在 2007/3/28 的配送路线:

配送点的需求矩阵, 如表VIII所示:

表Ⅷ 2007/3/28 的需求矩阵表

货物	V_3	V_5	V_6
货物 1 (Q1)	0	0	0
货物 2 (Q2)	0	0	0
货物 3 (Q3)	0	0	0
货物 4 (Q4)	5	10	14
货物 5 (Q5)	8	10	10

配送点 V_3 调用车辆的编号为：14；

配送点 V_5 调用车辆的编号为：3；

配送点 V_6 调用车辆的编号为：11，12；

空载率：0.064

经过一轮的配送，可以使车辆的平均空载率降低到 85%到 95%之间，充分利用的装载空间，使货物在交货时间内及时送达客户手中，提高配送的服务质量，并使车辆的调度更加合理，是标准化运作流程中简化操作的有用工具。



物畅其流 掌控自如