



用于解决集群车辆路径问题的混合粒子群优化算法Md. Anisul Islam a,*, Yuvraj Gajpal



b, Tarek Y. ElMekkawy c

^a 机械工程系, Room E2-327, EITC Building, University of Manitoba, 75A Chancellor Drive, Winnipeg, Manitoba, R3T 5V6, 加拿大

供应链管理系, 631-181 Freedman Crescent, 曼尼托巴大学阿斯珀商学院, 温尼伯, 马尼托巴省, R3T 5V4, 加拿大

^c 卡塔尔大学机械与工业工程系, 邮政信箱 2713, 多哈, 卡塔尔

文章信息

抽象的

文章历史:

2019年9月18日收到
2021年6月21日收到修订版
2021年6月23日接受
2021年6月30日上线

关键词:

集群车辆路径问题 (CluVRP)
粒子群优化 (PSO)

混合元启发式

本文考虑了经典的有能力车辆路径问题的一种变体,称为集群车辆路径问题 (CluVRP)。在 CluVRP 中,客户被分为不同的集群。在同一集群中的所有客户都得到服务之前,访问集群的车辆无法离开集群。

每个集群和客户只能被服务一次。针对具体问题,提出了一种新的混合元启发法,结合粒子群优化 (PSO)和可变邻域搜索 (VNS)来解决CluVRP。在混合PSO中,PSO的基本原理保证了解的多样性,VNS保证了解的强度,使解达到局部最优。我们对从 CluVRP 文献中获得的各种大小的基准实例进行了广泛的计算实验,以评估所提出的混合 PSO 的性能。将所提出算法获得的结果与文献中的结果进行比较,以验证所提出的混合 PSO 的有效性。事实证明,所提出的算法优于 CluVRP 上最先进的算法。所提出的算法在 293 个基准实例中获得了 138 个新的最著名的解决方案。

© 2021 Elsevier BV 保留所有权利。

一、简介

典型的车辆路径问题 (VRP)是一个物流分配问题。VRP 旨在获取在各种供需约束下为许多地理位置分散的客户提供成本最低的车辆路线列表。这是一个组合优化问题,需要指数级的计算时间来优化。本研究提出了容量车辆路径问题 (CVRP) 的一种变体,称为集群 VRP (CluVRP)。在 CluVRP 中,客户被划分为称为集群的预定义组。单个集群对应的客户在离开集群之前必须都被同一辆车访问过。VRP 中的聚类概念因其经济影响及其在建模和解决大量实际应用中降低的复杂性而广为人知[1]。

CluVRP 是 CVRP 的通用形式。由于 CVRP 被证明是一个 NP 难问题,因此 CluVRP 也是 NP 难问题[2]。

* 通讯作者。

电子邮件地址: islamma@myumanitoba.ca (MA Islam),
Yuvraj.Gajpal@umanitoba.ca (Y.Gajpal), tmekkawy@qu.edu.qa
(TY·埃尔梅卡维)。

<https://doi.org/10.1016/j.asoc.2021.107655>

1568-4946/© 2021 Elsevier BV 保留所有权利。

CluVRP 有两种变体,例如具有强集群约束的 CluVRP (CluVRP) 和具有弱集群约束的 CluVRP (SoftCluVRP)。在 CluVRP 中,属于同一集群的所有客户必须由同一辆车不间断地访问。

车辆在为顾客服务时不得多次进出集群。在 SoftCluVRP 中,虽然属于特定集群的客户由同一辆车访问,但车辆在路线上的行程中允许多次离开和进入集群。本文研究了一种具有强集群约束的 CluVRP,简称 CluVRP。许多研究都对 CluVRP 进行了探索,例如[1,3-10],而 SoftCluVRP 在 [8,10,11] 的作品中进行了研究。文献中的大多数研究提出了基于元启发式的解决方法。

Sevaux 和 Sörensen [12]引入的综合 CluVRP 重点关注快递公司的现实包裹递送问题,托运包裹按照具体投递区域分别放入对应的垃圾箱内。收货人属于指定为集群的同一区域。CluVRP 还可以出现在许多场景中,例如当客户喜欢与朋友或邻居搬家时运送老年人、为封闭社区提供服务、收集城市固体废物、提供普通维修人员的服务、送货

医疗保健提供服务和在紧急情况下和物流操作中优先订购[1,13]。

本文的主要贡献是设计一种混合元启发式方法来求解 CluVRP。所提出的元启发式算法基于粒子群优化 (PSO) 和 CluVRP 特定变量邻域搜索 (VNS) 的组合。VNS 有助于发现搜索区域的局部最优解。在文献中,VNS 主要是为了改进粒子解决方案而实施的。但是,本文使用 VNS 通过改进方案来改进个人最佳解决方案以及全局最佳解决方案。该贡献还包括 PSO 算法中新功能的使用,例如使用两种类型的粒子。这种混合 PSO 的目标是为 CluVRP 问题获得更好质量的解决方案。

本文的其余部分的结构如本文所述。第 2 节回顾了 CluVRP 的文献。第 3 节定义了 CluVRP,并提出了其数学公式。

所提出的混合 PSO 将在第 4 节中详细讨论。
计算结果在第 5 节中报告。最后,结论在第 6 节中陈述。

2.文献综述

Sevaux 和 Sörensen [12]为一家著名快递服务公司的配送操作提出了 CluVRP 的混合整数线性规划公式。巴泰勒米等人。 [3]为CluVRP设计了一种启发式算法,在所有簇间边缘上添加一个大值,将CluVRP转换为CVRP,并通过模拟退火方法求解。波普等人。 [4]提出了两种基于整数规划的 CluVRP 精确求解方法。在另一项研究中,Battarra 等人基于整数规划公式,提出了两种精确求解方法,例如分支截割法和分支截定价法。 [5]。 Marc 等人开发了一种基于遗传算法与模拟退火相结合的新混合算法来求解 CluVRP。 [6]。维达尔等人。 [7]提出了两种混合元启发式方法来求解 CluVRP。第一个是基于 Subra-manian [14] 设计的迭代局部搜索 (ILS)算法。第二个是基于统一混合遗传搜索 (UHGS) 。 Expósito-Izquierdo 等人提出了一种近似两级优化技术来求解 CluVRP。 [1]。 Defryn 和 Sörensen [8]开发了一种高效的两级变量邻域搜索 (VNS) 启发式方法来求解 CluVRP。

Pop 等人的一项研究。 [9]提出了一种独特的两级优化方法来解决 CluVRP。该问题分为两个子问题:上层 (集群)子问题和下层 (客户)子问题。该方法通过遗传算法获得访问集群的路径,然后通过Concorde TSP求解器确定客户在集群内的访问顺序。元启发式的最新趋势表明了其性能改进的混合作。最近,Hintsch 和 Irnich [10]为 CluVRP 提出了一种基于大型多邻域搜索 (LMNS) 的元启发式算法。该问题被分解为三个子问题:将集群分配给路由、集群内路由以及集群路由。在LMNS方法中,首先对簇使用多次破坏和修复动作,然后采用基于VND的局部搜索改进方案进一步优化。大多数杂交是通过使用局部搜索方案来完成的。

这一观察结果促使我们混合 PSO 以提高其在本研究中的性能。

我们当前的论文提出了一种基于经典粒子群优化 (PSO) 与可变邻域搜索 (VNS) 相结合的解决方案来求解 CluVRP。 PSO 是一种基于群体的组合优化技术,最初由 Eberhart 和 Kennedy [15] 所熟悉。技术

受到许多自然群体中的社会集体为的启发,例如鸟群、鱼群和人类。混合 PSO 方法被用于 VRP 的许多变体中,例如与 Ai 和 Kachitvichyanukul 中的局部搜索混合[16];在 Marinakis 等人中使用本地搜索和路径重新链接策略。 [17]并在 Norouzi 等人中修改了本地搜索。 [18]。此外,还建立了自适应 PSO 算法来成功解决集成码头起重机和堆场卡车调度问题[19]。德里迪等人。 [20]开发了一种新的基于 PSO 的解决方案,用于解决具有时间窗口和多车辆的多仓库取货和送货问题的优化问题。从文献中可以清楚地看出,PSO 的效率可以通过其混合来提高。

与其他元启发式算法相比,PSO 算法具有许多优点,例如需要调整的参数少,易于实现,并且需要更少的服务器内存。 PSO 算法已被成功利用,并被认为是运输、制造和调度问题领域许多组合优化问题的有效解决方法[21-23]。 VNS使用多种局部搜索方法来获得局部最优值。 PSO 具有使解决方案多样化的能力,而 VNS 具有强化解决方案的能力。这些优点结合在我们提出的元启发式算法中。

可变邻域搜索 (VNS)最初由 Mladenovic 和 Hansen [24]于 1997 年提出来解决商旅问题。通常,VNS 用作局部搜索算法来获得局部最佳解[25]。 VNS 也是 VRP 中广泛使用的启发式搜索方法[26]。许多研究发现使用 VNS 和 PSO 来解决多个优化问题,其中 PSO 解决方案用作全局搜索算法。

马里纳斯基等人。 [27]通过从贪婪随机自适应搜索过程生成初始解决方案并通过 VNS 算法进一步改进解决方案,生成了混合 PSO 元启发式算法来求解 CVRP。戈克萨尔等人。 [28]引入了一种基于PSO和可变邻域下降 (VND) (一种较低级别的VNS)的混合元启发式算法,来解决同时取货和送货的车辆路径问题。此外,马里纳斯基等人。 [29]针对具有时间窗的车辆路径问题提出了一种多自适应 PSO 解决方案,其中通过对群体中的每个粒子应用 VNS 来改进 PSO 解决方案。邹等人。 [30]提出了一种与 VNS 混合的新型 PSO 算法,用于解决具有时间窗的取货和送货问题的多目标 VRP。张等人。 [31]设计了一种基于 VNS 与二进制 PSO 集成的混合解决方案来解决位置路由问题 (LRP) 。 Marinakis [32]将 PSO 与 VNS 相结合来求解有能力的 LRP。在另一项研究中,Moghaddam 等人。 [33]在基于 PSO 的先进解决方案中使用 VNS 来解决需求不确定的车辆路径问题。使用一种新颖的解码算法来提高解决方法的效率。解码旨在生成车辆路线和更新粒子值。此外,由于PSO在产生强大的全局解决方案方面的主导为,而VNS具有生成最佳局部解决方案的优势,PSO和VNS也被广泛应用于作业调度问题[34]。刘等人。 [35]使用基于PSO与VNS相结合的混合元启发式方法来解决多目标灵活作业车间调度问题。在其他工作[36] 中,结果表明,对于作业车间调度问题,不与 PSO 混合的更简单的 VNS 算法比带有 VNS 算法的混合 PSO 能以更短的 CPU 时间提供更好质量的解决方案。此外,Ali 等人提出了一种结合 PSO 和 VNS 算法的混合元启发式算法来解决无约束全局优化问题。 [37]。研究中采用PSO对解空间进行更广泛的多样化和深度强化,并采用VNS作为局部搜索算法。此外,基于 PSO 的混合元启发式算法被设计用于排列流

车间调度问题[38]。在工作中,PSO 算法是与随机 VNS 合并,提出了 VNS 的一种变体在[32]中,与模拟退火混合以增强 PSO 在解决方案中的探索能力。古迈达和 Luo [39]开发了一种新的基于混合优化技术 PSO 与 VNS 相结合以增强定位过程在无线传感器网络中。马里纳基斯等人。 [40]设计了一个混合 PSO 与 VNS 结合解决约束最短问题路径问题。蔡等人。 [41]提出了一种基于混合 PSO 的解决方案 PSO 与 VNS 混合来求解 VRP 的方法通过减少碳排放来调整速度变量路线。研究了铁路货物运输问题 Nie 中提出了一种基于 PSO 和 VNS 的求解方法等人。 [21]。Ranjbar 和 Saber [42]设计了 VNS 并进了修改基于 PSO 的转运调度解决方案单站多产品问题。伊斯兰等人。 [43]提出了一种基于 PSO 和 VNS 的解决方案来解决碳排放上限下的混合车队绿色物流问题。受这一观察的启发,本文将 VNS 嵌入 PSO 以获得高质量的 CluVRP 解。

3.CluVRP的问题定义

CluVRP 可以定义在无向图G = (V, E) 上,其中V = {0, 1, 2, . . . , n},一组节点 (顶点) ,包括客户{1, 2, . . . , n}, E是连接每对的弧的集合 V 中的节点(i, j)和车库 0。同质车队位于车辆段,车辆开始和结束的地方在为顾客服务的同时进旅。

参数	
n	客户总数
	簇总数
0 _	车厂
米	第 l 个集群的客户数量
-	个人车辆
中号	网络中可用的车辆总数
r	单个簇 (互斥非空不相交) , r ∈ R
博	集群组
土 _	集群的需求, r (汇总所有集群中的客户) , dr > 0
tcij	从 i 到 j 的边的非负旅成本 j,(i, j) ∈ E
问 铬	每辆车最大装载能力, Q > 0 集群内的客户群, Cr = {i ∈ n:ri = r}, ∀ r ∈ R
在	顶点集
S	客户节点的任意子集{1, 2, . . . , n}
δ +(S)	边集(i, j) ,其中i ∈ S且j ∈ V \ S
δ -(S)	边集(i, j) ,其中i ∈ V \ S且j ∈ S

二元决策变量是:

否则为 0, $x_{ijm} = \begin{cases} 1 & \text{车辆 } m \text{ 从客户 } i \text{ 驶到 } j \\ 0 & \text{否则} \end{cases}$

否则为 0, $y_{im} = \begin{cases} 1 & \text{名称 } i \text{ 由车辆 } m \text{ 服务} \\ 0 & \text{否则} \end{cases}$

CluVRP 可以表述如下:

最小化 $\sum_{(i,j) \in E} \sum_{m=1}^M tc_{ij} x_{ijm}$ (1)

英石, $\sum_{m=1}^M \varnothing = 1 \quad \forall i \in \{1, 2, n\}$ (2)

$\sum_{m=1}^M y_{0m} \leq M$ (3)

$y_{0m} \geq 8 \quad \forall m \in \{1, 2, M\}, \forall i \in \{1, 2, n\}$ (4)

$\sum_{j=1}^n x_{ijm} = \sum_{j=1}^n x_{jim} = \varnothing \quad \forall m \in \{1, 2, M\}, \forall i \in \{0, 1, 2, n\}$ (5)

$\sum_{i=0}^n \text{说} \leq Q \quad \forall m \in \{1, 2, M\}$ (6)

$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ijm} \geq y_{hm} \quad \forall S \subseteq \{1, 2, n\}, h \in S, m \in \{0, 1, 2, M\}$ (7)

$\sum_{(i,j) \in \delta^+(Cr)} \sum_{m=1}^M x_{ijm} = \sum_{(i,j) \in \delta^-(Cr)} \sum_{m=1}^M x_{ijm} = 1 \quad \forall r \in R$ (8)

$\sum_{i=1}^n \text{说} \geq \sum_{i=1}^n \text{说} + 1 \quad \forall m \in \{1, 2, M - 1\}$ (9)

$x_{ijm} \in \{0, 1\} \quad \forall (i, j) \in E, \forall m \in \{1, 2, \dots\dots\dots M\}$ (10)

$y_{im} \in \{0, 1\} \quad \forall i \in \{0, 1, 2, n\}, \forall m \in \{1, 2, M\}$ (11)

确定了最小化总旅成本的目标
由方程式(1)约束 (2)保证每个客户都被访问过正好一次。约束 (3)保证车辆数量使用的车辆数量不超过可用车辆的数量。约束 (4)强制执路线中每辆车都应该访问的规则仓库。如果车辆m不访问停车场,那么它应该不拜访任何客户。约束 (5)确保到达对于给定的客户,出发车辆是相同的。约束 (6)规定了车辆的最大装载能力

很满意。约束(7)表示子游消除约束。约束(8)保证每个簇都可以被访问恰好由一辆独特的车辆一次。约束条件(9)是不等式确保部分对称。

4. 为 CluVRP 提出的混合 PSO

所提出的方法是一种混合 PSO 算法,结合了标准 PSO 和 VNS。VNS 的结构

所提出的方法受到 Vidal 等人的一项研究的启发。 [7]。一般来说,PSO 的性能很大程度上受问题映射的准确性。因此,PSO 被修改为符合本研究中的问题规范。主要的所提出的混合 PSO 的特点是使用两种类型代表集群和客户的粒子,以及使用个人最佳解决方案的改进方案。伪所提出的混合 PSO 的代码如算法 1 所示。拟议的混合 PSO 使用以下定义:

α_i	第 l 个粒子中第 i 个粒子的当前簇位置值方面
v_{ij}	第 i 个粒子的当前客户位置值第 j 维
β_{il}	第 l 个粒子当前的簇速度值方面
δ_{ij}	当前第 i 个粒子的客户速度值第 j 维
是	粒子的适应度函数, i

$A_{i,j}^z$	迄今为止找到的个人最佳簇位置值
C_{ij}^z	对于第 i 维的第 j 个粒子找到的个人最佳客户定位值
F_i^z	到目前为止,对于第 j 个维度的第 i 个粒子最佳粒子的适应度函数
A_i^*	第 i 个找到的全局最佳簇位置值方面
C_j^*	全局最佳客户地位价值第 j 维
fg_{in}	全局最佳粒子的适应度函数惯性系数
c_1	认知系数
c_2	社会系数
r_1, r_2	独立随机数
K	粒子总数
X	顾客群的位置矩阵
和	簇群的位置矩阵
在	顾客群速度矩阵
\otimes	集群群的速度矩阵
X^G/X^G	客户个人最佳/全局最佳位置值对于群体
和 z /和 G	集群个人最佳/全局最佳位置值一群
S^z	群体的个人最佳解决方案

Algorithm 1: Pseudo code of the proposed algorithm

```
1: Initialization
2: Set parameters:  $w = 0.7, c_1 = c_2 = 2, r_1 = r_2 = 0.5, K = n/4$ .
3: Initialize position matrix  $X, Y$  and velocity matrix  $U, V$ 
4: Initialize the personal best fitness vector  $f^b$ 
5: Initialize the global best fitness vector  $f^g$ 
6: Main phase
7: Do while
8:      $S \leftarrow \text{GenerateCluVRPSolution}(X, Y, U, V)$ 
9:      $S \leftarrow \text{VNS}(S)$ 
10:    Update personal best matrix  $X^b, Y^b$ , fitness vector  $f^b$ , and personal best solution matrix  $S^b$ 
11:    Improve personal best matrix using improvement scheme
12:     $S^b \leftarrow \text{Improvement scheme}(S^b)$ 
13:    Update the best particle  $X^G, Y^G$  and fitness vector  $f^g$ 
14:    Update  $(X, Y, U, V)$ 
15: End Do
```

4.1.初始化阶段

位置和速度向量初始化如下:

$$a_{il} = a_{min} + (a_{max} - a_{min}) * U(0, 1)$$
$$\forall i \in \{1, 2, \dots, K\}, \forall l \in \{1, 2, \dots, C\} \tag{12}$$
$$y_{il} = y_{min} + (y_{max} - y_{min}) * U(0, 1)$$
$$\forall i \in \{1, 2, \dots, K\}, \forall j \in \{1, 2, \dots, \text{的}\} \tag{13}$$
$$\delta_{il} = \delta_{min} + (\delta_{max} - \delta_{min}) * U(0, 1)$$
$$\forall i \in \{1, 2, \dots, K\}, \forall l \in \{1, 2, \dots, C\} \tag{14}$$
$$\beta_{il} = \beta_{min} + (\beta_{max} - \beta_{min}) * U(0, 1)$$
$$\forall i \in \{1, 2, \dots, K\}, \forall j \in \{1, 2, \dots, \text{的}\} \tag{15}$$

其中 $a_{max} = y_{max} = \delta_{max} = \beta_{max} = 4$; $a_{min} = y_{min} = \delta_{min} = \beta_{min} = -4$ 。

这里, $U(0, 1)$ 表示生成的均匀随机数0到1之间。粒子的个人最佳适应度向量,和全局粒子的适应度向量被初始化为无穷大。

$$F_i^z = \infty \forall i \in \{1, 2, \dots, K\}$$
$$F^G = \text{无穷大}$$

表格1

具有 6 个集群的实例,其位置值和需求为任意迭代, t_0 。

集群	1	2	3	4	5	6
位置值, a_{il}	1.99	3.67	-2.25	2.50	-0.09	1.08
集群需求,博士	45	10	25	15	25	30

4.2.映射位置向量以生成 CluVRP 解决方案

PSO 通常将粒子的位置值映射为生成给定问题的解决方案。位置值用于生成 CluVRP 解 ($S \leftarrow \text{GenerateCluVRP-Solution}(X, Y, U, V)$),如算法 1 第 8 所述。

许多研究使用两阶段方法来生成 CluVRP 解决方案[8,9]。在所提出的 PSO 中,解决方案是在两个阶段。第一阶段,车辆集群路线是根据簇 a_{il} 的位置值生成的,而每个集群的客户路由在第二阶段生成根据客户 y_{ij} 的位置值。

4.2.1.生成集群路由

集群路由的生成从空程开始对于每辆车,车辆开始和结束程的位置仓库。集群被迭代地添加到车辆路线中找到完整的解决方案。首先,最高的簇选择位置值以包含在车辆路线中,然后将所选集群插入到车辆路线中使用最便宜的插入方法。然而,簇插入可能会面临车辆没有足够运力的情况用于插入选定的簇。在这种情况下,禁忌搜索使用基于搜索的方法来插入所选簇。这种方法尝试使用最大化可用车辆容量交换 (1,1) 和移位 (1,0) 邻域移动。选定的交换

对于下一个 $U(c_{ij})$ 迭代。类似地,在 $\text{shift}(1,0)$ 移动中,簇的插入是下一个 $U(c^*v)$ 在簇中被禁止 $\frac{2}{8}, \frac{c_2}{4}$ 迭代。

要了解集群路线的映射过程,请考虑一个具有 6 个集群和 2 辆车的实例,其中有 1 辆车

容量为 80。在任何迭代 t 中,考虑以下集群第 i 个粒子在第 i 维的位置值如表1所示。在这个例子中,6 个不同的维度代表 6 个不同的维度。集群。由于不同的维度与不同的簇,我们将第 i 维的簇位置值称为第 i 个簇的位置值。

在映射中,簇按非递增顺序排列他们的位置值。所得顺序为 $\pi = 2-4-1-6-5-3$ 。两条车辆路线最初从前两个集群开始从 π 。初始路线为 $\{0-2-0; 0-4-0\}$ 以及剩余的每辆车的车辆容量都会相应更新。然后,选择集群 1 插入车辆路线。插入集群 1 的成本 (即总路线长度的增加)的评估为两条路线的每个位置 $\{0-2-0; 0-4-0\}$ 。假设最便宜的簇 1 的插入是通过在簇 3 的位置 3 处插入来获得的车辆 2。则新路线为 $\{0-2-0; 0-4-1-0\}$ 。下一个迭代中,选择簇 6 进插入。假设最便宜的簇 6 的插入是通过在簇 3 的位置 3 处插入来获得的车辆 1。则新路线为 $\{0-2-6-0; 0-4-1-0\}$ 。下一个迭代中,选择簇 5 进插入。假设最便宜的簇 5 的插入是通过在簇 2 的位置 2 处插入来获得的车辆 1。则新路线为 $\{0-5-2-6-0; 0-4-1-0\}$ 。在这点,两辆车的剩余容量分别为 15 和 20。但是未分配集群 3 的需求是 25 并且没有车辆具有容纳集群 3 所需的容量。在这种情况下,我们使用带有 swap (1, 1) 和 shift (1, 0) 的禁忌搜索

目标函数是最大化剩余车辆

表2
2个集群的车辆路线,其客户和位置值位于任意位置迭代, t_0 .

集群1	顾客	10	7
	位置值, y_{ij}	2.74	-1.81
集群3	顾客	5	
	位置值, y_{ij}	2 2.03	17 -0.96

容量。当目标函数达到目标函数时,禁忌搜索停止 (即剩余车辆容量)至少变为 25。假设禁忌搜索发现新路线为 {0-4-5-2-6-0; 0-1-0}。这车辆 1 和车辆 2 的剩余容量分别为 0 和 35 分别。最后,选择簇 3 进插入。假设通过在位置插入获得簇 3 最便宜的插入车辆 2 上的路线为 3。因此,最终路线为 {0-4-5-2-6-0; 0-1-3-0}。

4.2.2.生成客户路线

一旦构建了集群路由,就会产生一系列生成每个集群的客户以找到 CluVRP 的完整解决方案。生成客户的序列

通过选择与集群相似的客户路线生成方法见4.2.1 节。

要了解客户路线的生成,请考虑车辆中的集群路线是{0-1-3-0}。假设有 3 个集群 1 和集群 3 中分别有 4 个客户和 4 个客户如表2所示。在任何迭代 t 中,请考虑以下因素第 i 个粒子在第 j 维中的客户位置值如下如表2所示。由于不同的维度与不同的客户,我们参考第 j 维的位置值作为第 j 个客户的位置值。
在客户路线生成中,对客户进排列按其位置值的非递增顺序。由此产生的集群 1 的客户订单为 $\tau = 4 - 10 - 7$, 集群 3 的客户订单为 $\tau = 4 - 10 - 7$ 是 $\tau = 2 - 5 - 9 - 17$ 。车辆完整的客户路线是 {0-4-10-7-2-5-9-17-0}。旅成本 (即目标函数路线的价值)是所有客户的旅成本之和在路线中。

4.3. CluVRP 的变量邻域搜索 (VNS)

所提出的 PSO 将位置向量视为一个区域而不是特定的点。映射中生成的解决方案

阶段代表该地区的一种解决方案,这可能不是该地区的最佳解决方案。因此,采用VNS达到局部最优。 VNS 程序包括三个步骤本地搜索动作,包括路由间搜索、路由内搜索和簇内搜索。路线间搜索和路由内搜索集中在集群级别;而集群内搜索则侧重于客户级别。邻里关系

在集群级别使用的运算符:shift,shift2,swap、路由间搜索中的swap(2,1)、swap(2,2)和2-opt;和路由内搜索中的 shift-or-opt2,or-opt3,2-opt 和 swap。NLc是所有路由间邻域搜索的列表。这用于簇内搜索的邻域算子 (客户级别)有shift,2-opt,swap;这些探索所有在每个簇内移动。运营商的详细信息可以见于文献[7,14,44]。各个算子的结构如图所示。 1和2. 第一步采用策略是适用于所有本地搜索动作。在该策略中,解决方案只要找到改进的解决方案就会更新。在所有本地搜索中,每个邻域移动仅选择一次

可能的改进而不是迭代策略。整体VNS的结构如算法2所示。

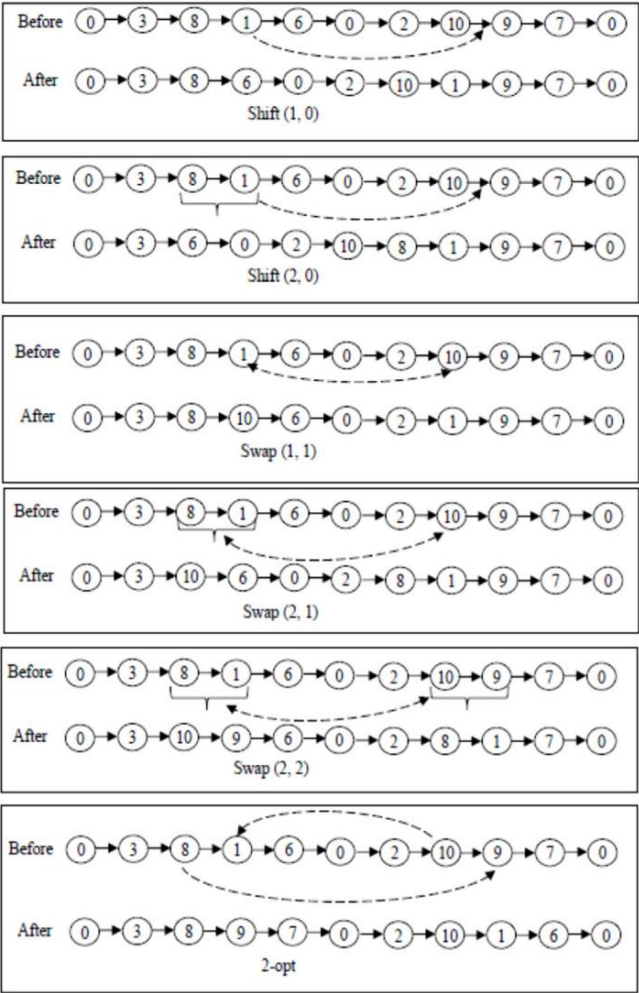


图 1.路线间邻域搜索算子。

Algorithm 2: Variable neighborhood search (VNS)

```
1: Method VNS:
2: Initial solution,  $s$ ;
3: Do
4:   Set previous solution,  $s^{initial} = s$ ;
5:   List ( $NL_c$ ) for the inter-route search;
6:   While  $NL_c \neq \emptyset$ 
7:     Choose randomly a neighborhood from  $NL_c$ ;
8:     Find best  $s^*$  of  $s \in \text{neighbourhood}$ ;
9:     if  $f(s^*) < f(s)$ 
10:       $s \leftarrow s^*$ ;
11:       $s \leftarrow \text{Intra-route search}(s)$ 
12:      Update  $NL_c$ ;
13:     else
14:       Remove neighbourhood from  $NL_c$ ;
15:   end While
16:    $s \leftarrow \text{Intra-cluster search}(s)$ ;
17: While ( $s < s^{initial}$ )
18:   return  $s$ ;
19: end VNS;
```

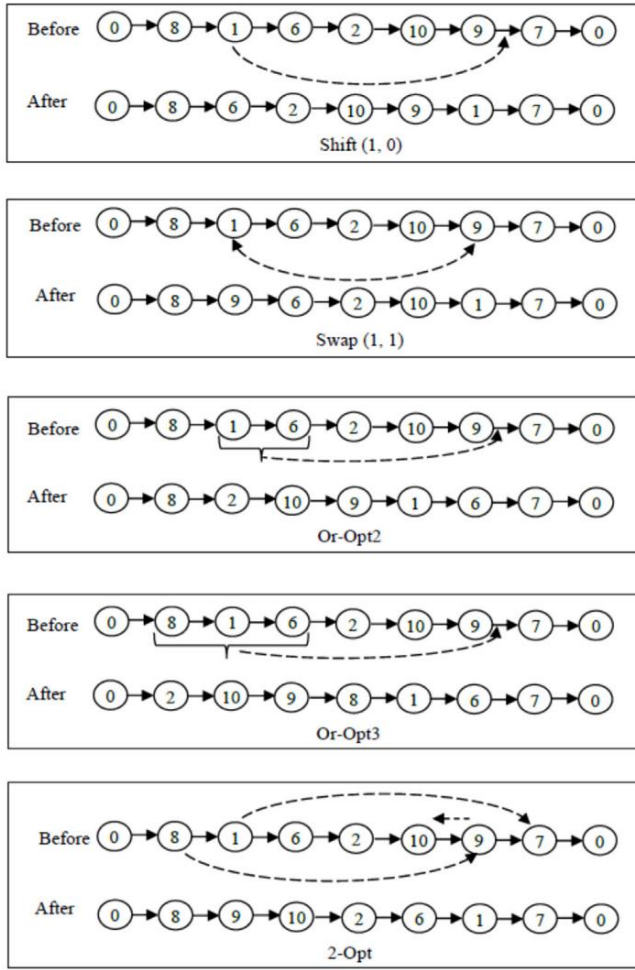


图 2.路由内和簇间邻域搜索算子。

4.4.更新位置和速度矢量

如果当前获得的解优于当前个人最佳解,则更新每个粒子的个人最佳位置值。类似地,如果发现新的最佳解决方案比当前的全局最佳值更好,则更新全局最佳值。

速度和位置向量更新如下: $-ail) + c2r2(\alpha$

$$\delta il = w\delta il + c1r1(a_i^p - ail) \quad \forall i \in \{1, 2, \dots, K\}, \forall l \in \{1, 2, \dots, C\} \quad (16)$$

$$\beta il = w\beta il + c1r1(c_j^p - \beta il) + c2r2(c_j^* - \beta il) \quad \forall i \in \{1, 2, \dots, K\}, \forall j \in \{1, 2, \dots, n\} \quad (17)$$

$$ail = ail + \delta il \quad \forall i \in \{1, 2, \dots, K\}, \forall l \in \{1, 2, \dots, C\} \quad (18)$$

$$\beta il = \beta il + \beta il \quad \forall i \in \{1, 2, \dots, K\}, \forall j \in \{1, 2, \dots, n\} \quad (19)$$

4.5.改善方案

改进方案用于改进个人最佳解决方案。这是本研究使用的 PSO 的新特征之一。据我们所知,现有的 PSO 文献中并没有使用这个特征。在改进方案中,首先对解进行扰动,生成新的解。然后使用 VNS 方案对扰动解进行优化。在集群和客户级别都实施了扰动技术。

在扰动方案中,首先删除 $\Delta 1/\Delta 2$ 数量的集群/客户,然后使用

最便宜的插入方法。改进方案的结构如算法3所示。参数 $\Delta 1$ 和 $\Delta 2$ 分别在 $[0.5c, 0.75c]$ 和 $[0.5nl, 0.75nl]$ 之间随机生成。

Algorithm 3: Improvement scheme

```

1: Method Improvement scheme:
2:   Initial solution,  $s$ ;
3:    $s^* \leftarrow$  Perturbation ( $s$ )
4:    $s^{**} \leftarrow$  VNS ( $s^*$ )
5:   Update  $s$ 
6:   if  $f(s^{**}) < f(s)$ 
7:      $s = s^{**}$ 
8:   return  $s$ ;
9: end Improvement scheme;

```

4.6.混合 PSO 的计算复杂度

混合 PSO 算法有四个主要步骤 (1)序列生成, (2)VNS方法, (3)参数更新和 (4)改进方案。序列生成步骤首先为集群创建路由。聚类路线生成执行两个连续操作 - (a) 根据位置值排列聚类, 以及 (b) 在部分生成的路线中插入聚类。这两个操作都可以在 $O(c^2)$ 时间内完成, 集群路由生成步骤的复杂度仍然是 $O(c^2)$ 。生成集群路由时, 序列生成步骤创建客户的路由, 这可以在 $O(n^2)$ 时间内执行。

后

由于集群路径生成和客户路径生成是顺序执行的, 因此顺序生成步骤的复杂度变为 $O(c^2 + n^2)$ 。类似地, 可以进入 VNS 方法、参数更新和改进方案

在 $O(c^2)$ 时间内。PSO 的四个步骤是顺序执行的, 因此混合 PSO 的一次迭代的复杂度仍然是 $O(c^2 + n^2)$ 。

5. 计算实验

所提出的混合 PSO 算法是使用 C++ 编程语言来实现的, 以求解 CluVRP 文献中的几个基准数据集。实验在 Linux 服务器上运行, 该服务器配备四个 2.1 GHz 处理器, 每个处理器 16 核, 总共 256 GB RAM。

5.1.基准 CluVRP 实例

混合 PSO 的性能在 CluVRP 基准实例上测试, 该实例由 20 个主要客户组组成, 分别命名为 A、B、P、M 和黄金实例 (黄金 1 至黄金 20), 总共 298 个单独实例。这些 CluVRP 实例最初由 Bektas 等人从 GVRP 实例中采用。[45]。表3总结了基准数据集的特征。表4显示了本研究中用于结果报告目的的算法及其符号。

通过使用集合 A、B、M 和 P 的问题实例进行敏感性分析来设置 PSO 参数。我们使用没有 VNS 且没有改进方案的 PSO 解决方案进行 100 次迭代来设置参数。我们从文献 [16, 17, 27] 中找到的参数值开始敏感性分析。参数值按照 w 、 $c1$ 、 $c2$ 、 $r1$ 、 $r2$ 、 K 的顺序——设置。每个参数的多个不同替代值被测试为 $w = \{0.5, 2\}$; $c1 = \{2, 5\}$; $c2 = \{2, 5\}$;

表3

基准实例的摘要。

实例类型	数量实例	数量顾客	数量集群	车辆容量(车辆数量)	来源
A	27	31-79	11-27	100 (2-5)	贝克斯等人。 [45]
乙	23	30-77	11-23	100 (2-5)	贝克斯等人。 [45]
中号	4	100-261	34-76	200 (3-8)	贝克斯等人。 [45]
磷	24	15-100	6-51	35-400 (1-8)	贝克斯等人。 [45]
金的	220	201-483	17-97	550-1000 (4-12)	巴塔拉等人。 [5]

表4

本研究中使用的算法及其符号。

符号	算法
公司前	Battarra 等人的分支和切割方法。 [5]
超高质量测序	Vidal 等人的统一混合遗传搜索方法。 [7]
两级	Exposito-Left 等人的二级算法结果。 [1]
两级迷走神经刺激	Defryn和Sorensen的二级变量邻域搜索结果 [8]
基于分解的方法	Horvat-Marc 等人的分解方法。 [6]
两级优化	Pop 等人的两级优化方法。 [9]
LMNS	Hintsch 和 Irnich 的大型多重邻域搜索结果 [10]
混合粒子群算法	本文提出的算法

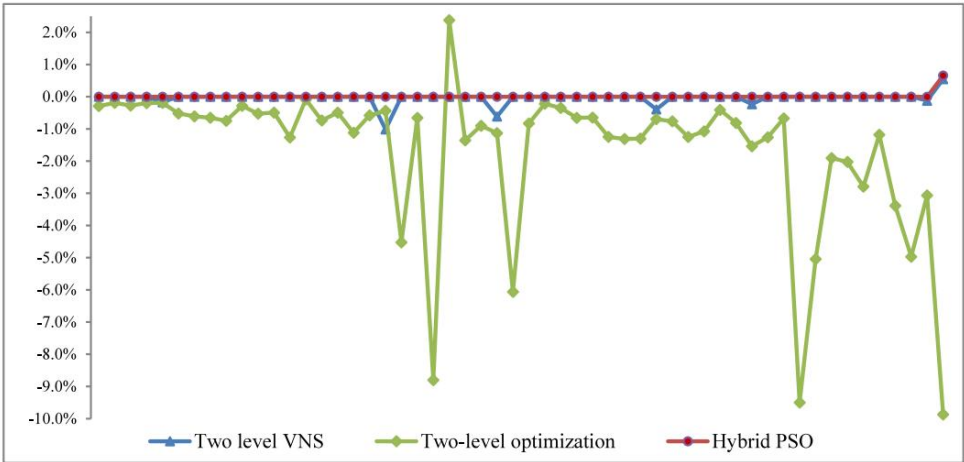


图 3. A,B,M 实例的算法结果的改进百分比。

$r1 = \{0, 1\}$; $r2 = \{0, 1\}$ 。最后我们将最佳参数设置为 $w = 0.7$; $c1 = c2 = 2$; $r1 = r2 = 0.5$; $K = n/4$ 。我们跑建议的混合 PSO 为每个实例十次,其中 100 迭代 (即算法终止标准)。最好的结果每个实例的平均 CPU 时间是通过十次运行获得的。我们观察到 100 次迭代后结果有所改善是非常边缘化的。

5.2.不同算法的性能评估

本研究中的所有结果都是通过比较 Battarra 等人报告的结果。[\[5\]](#)使用分支和剪切 (BC) 解决 CluVRP 问题的算法。他们不能获得所有问题实例的最佳解决方案,但是报告了在执其算法期间获得的最佳可上限解决方案。Battarra 的解决方案

等人。[\[5\]](#)用UB表示。总体而言,性能算法,包括从文献中获得的算法,是通过两个标准进行评估。第一个标准是有多上实例中算法是否找到了比上限, UB解。报告在下面的表格中不。改进的UB”。第二个标准是改进%该算法与UB 相比。它是通过等式测量的。[\(20\)](#),其中Sol用于表示其他人找到的解决方案算法。组实例的改进百分比报告为表中的“改进%”。另外,处理时间

(CPU 时间)报告为 t (s)。下面的公式用于计算 UB 的改进百分比。

改善% =
$$\frac{UB - 溶胶}{UB} \times 100$$
(20)

[表 5](#)和[表 6](#)显示了本研究的所有结果,包括文献报道的结果。在性能评估中,采用了统计检验、非参数Friedman检验和事后Bonferroni 检验检查性能是否存在显著差异算法。弗里德曼的检验仅揭示了两者之间的差异不同算法的结果。执 Bonferroni 测试经过弗里德曼的测试以显示哪对特定的算法比较起来各有不同[\[46\]](#)。统计软件IBM SPSS版本19用于运Friedman

和事后 Bonferroni 检验,使用 $\alpha = 0.05$ 作为水平意义。

5.2.1. A,B,M,P 实例的性能评估[表 5](#)报告了实例组 A,B,M 和 P,两级VNS算法,基于分解的方法,两级优化,并在混合 PSO 中进行评估桌子。比较表明,所有两级 VNS、基于分解的方法和两级优化为 75 个实例中的一个实例获取改进的UB解决方案实例;而混合 PSO 能够获得

表5
A、B、M 和 P 实例的汇总结果。

BC省实例		两级迷走神经刺激				基于分解的方法			两级优化			混合粒子群算法		
组号	实例	数量 顾客	改进数量 UB	改进 %	时间 (秒)	数量 改进的UB	改进 %	时间 (秒)	数量 改进的UB	改进 %	时间 (秒)	数量 改进的UB	改进 %	时间 (秒)
A	27	31-79	0/24	-0.07%	0.05		-2.6%	...		-1.21%	...	0	0.00%	0.06
乙	23	30-77	0	-0.03%	0.04	1	-3.0%	...	1	-1.63%	...	0	0.00%	0.04
米4		100-261		0.11%	3.48	0 0	-32.3%	...	0 0	-5.32%	...		0.09%	2.09
第24页		15-100	1	-0.01%	0.07		0.13%	0.27
总计 78	0 1/75	1/78	1/78	1 1 2/78
平均	-0.03%	0.23	...	-5.00%	-1.7%	0.05%	0.22



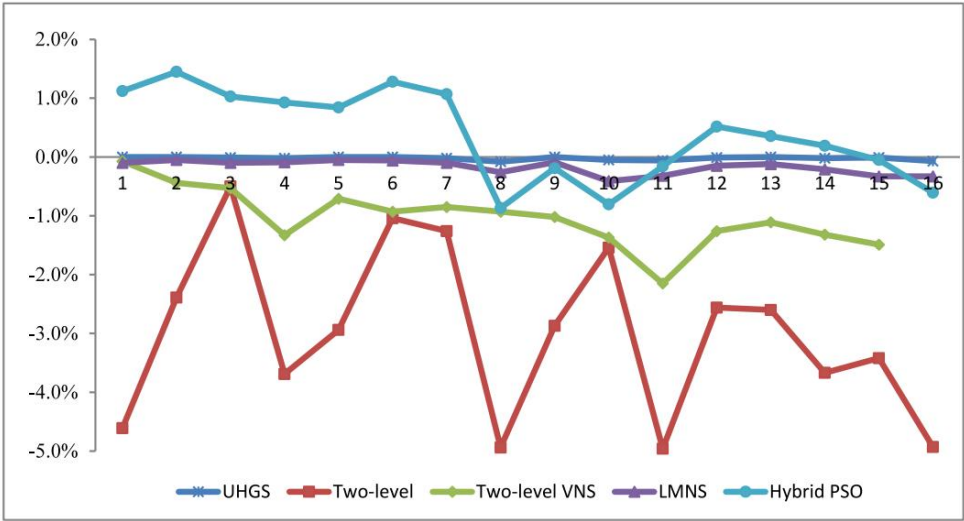


图 4. 16 组黄金实例的算法结果的改进百分比。

表7
杂交对溶液质量的影响。

混合程度 没有 VNS 和没有改进方案	迭代次数	改良UB数量	改进 %	时间 (秒)
的 PSO 有 VNS 和没有改进方案的 PSO 纯改进方案 提议的 PSO	3000	0	-83.59%	11.09
	350	0	-1.05%	10.44
	14000	94	0.01%	9.67
	100	138	0.31%	6.99

表8
实例 A,B 的结果。

实例	n	C	尿布_	混合粒子群算法		
				解决方案	CPU 时间 (秒)	改进 %
A	31	11	2	522	0.02	0.00%
A	32	11	2	472	0.04	0.00%
A	32	11	2	562	0.02	0.00%
A	33	12	2	547	0.03	0.00%
A	35	12	2	588	0.04	0.00%
A	36	13	2	569	0.04	0.00%
A	36	13	2	615	0.04	0.00%
A	37	13	2	507	0.04	0.00%
A	38	13	2	610	0.05	0.00%
A	38	13	2	613	0.06	0.00%
A	43	15	2	714	0.08	0.00%
A	44	15	3	712	0.07	0.00%
A	44	15	3	664	0.05	0.00%
A	45	16	3	664	0.08	0.00%
A	47	16	3	683	0.08	0.00%
A	52	18	3	651	0.09	0.00%
A	53	18	3	724	0.09	0.00%
A	54	19	3	653	0.08	0.00%
A	59	20	3	787	0.09	0.00%
A	60	21	4	682	0.08	0.00%
A	61	21	3	778	0.09	0.00%
A	62	21	4	801	0.08	0.00%
A	62	21	3	865	0.08	0.00%
A	63	22	3	773	0.07	0.00%
A	64	22	3	725	0.07	0.00%
A	68	23	3	814	0.08	0.00%
A	79	27	4	972	0.09	0.00%
Z	30	11	2	375	0.02	0.00%
Z	33	12	2	416	0.16	0.00%
Z	34	12	2	562	0.01	0.00%
Z	37	13	2	431	0.01	0.00%
Z	38	13	2	321	0.01	0.00%
Z	40	14	2	476	0.01	0.00%
Z	42	15	2	415	0.01	0.00%
Z	43	15	3	447	0.01	0.00%
Z	44	15	2	506	0.01	0.00%
Z	44	15	2	391	0.03	0.00%
Z	49	17	3	467	0.02	0.00%
Z	49	17	3	666	0.02	0.00%
Z	50	17	3	585	0.03	0.00%
Z	51	18	3	427	0.05	0.00%
Z	55	19	3	433	0.03	0.00%
Z	56	19	3	634	0.05	0.00%
Z	56	19	3	753	0.04	0.00%
Z	62	21	3	685	0.04	0.00%
Z	63	22	4	526	0.04	0.00%
Z	65	22	3	687	0.05	0.00%
Z	66	23	4	626	0.08	0.00%
Z	67	23	3	588	0.09	0.00%
Z	77	26	4	721	0.11	0.00%

表9
实例 M,P 的结果。

实例	n	C	尿布_	混合粒子群算法		
				解决方案	CPU 时间 (秒)	改进 %
中号	100	34	4	607	0.54	0.00%
中号	120	41	3	691	0.67	-0.29%
中号	150	51	4	804	2.52	0.00%
中号	199	67	6	914	9.08	+0.66%
大号	100	51	5	679	6.69	+3.18%
大号	15	6	4	253	2.53	0.01
大号	18	10	2	186	1.86	0.01
大号	19	7		200	2.00	0.01
大号	20	7		190	1.90	0.01
大号	21	8		202	2.02	0.01
大号	21	8	1	365	3.65	0.03
大号	22	8	1	279	2.79	0.02
大号	39	14	1	396	3.96	0.06
大号	44	15	4	440	4.40	0.09
大号	49	17	3	491	4.91	0.10
大号	49	17	2	447	4.47	0.10
大号	49	17	2	460	4.60	0.10
大号	50	17	4	537	5.37	0.13
大号	54	19	3	500	5.00	0.13
大号	54	19	3	595	4.71	0.23
大号	54	19	4	462	4.62	0.26
大号	54	19	4	471	5.95	0.17
大号	59	20	6	552	5.52	0.35
大号	59	20	3	611	6.11	0.21
大号	64	22	3	619	6.19	0.40
大号	69	24	4	643	6.43	0.47
大号	75	26	5	581	5.81	0.84
大号	75	26	4 4 2 2	581	5.81	0.84

六,结论

本文考虑组合优化问题 CluVRP。在CluVRP中,客户被分区到预定义的集群中。分配同一辆车来服务在移动到集群之前,所有客户连续位于集群下另一个集群或返回仓库。所有客户和集群只能服务一次。问题的目标是找到物流网络的最优配送成本使用可用车辆为所有客户提供服务。在这个论文提出了一种混合PSO算法来求解CluVRP。由于两种算法的互补性,混合算法PSO结合了VNS的局部最优改进能力具有 PSO 基于群体的多样化能力。这

表10
黄金实例 1-4 的结果。

实例				公共数据	混合粒子群算法		
团体	n	C	原布_		解决方案 CPU t (s) 4751	改进 %	
金 1 240 17 4 4831					4757	3.66	1.66%
金 1 240 18 4 4847					4789	2.42	1.86%
金 1 240 19 4 4872					4790	2.45	1.70%
金 1 240 21 4 4889					4826	2.57	2.02%
金色 1 240 22 4 4908					4818	2.58	1.67%
金色 1 240 25 4 4899					4862	2.61	1.65%
金 1 240 27 4 4934					4953	2.60	1.46%
金条 1 240 31 4 5050					5047	2.68	1.92%
金色 1 240 35 4 5102					5058	2.98	1.08%
金色 1 240 41 4 5097					4953	3.64	0.77%
金色 1 240 49 3 5000						4.38	0.94%
金2 320 22 4 7716					7622	6.10	1.22%
金2 320 23 4 7693					7578	6.04	1.49%
金2 320 25 4 7668					7571	6.14	1.26%
金2 320 27 4 7638					7527	5.27	1.45%
金2 320 30 4 7617					7552	4.55	0.85%
金2 320 33 4 7640					7548	4.12	1.20%
金2 320 36 4 7643					7550	4.71	1.22%
金2 320 41 4 7738					7644	4.80	1.21%
金2 320 46 4 7861					7795	5.59	0.84%
金2 320 54 4 7920					7830	7.27	1.14%
金2 320 65 4 7892					7841	10.32	0.65%
金3 400 27 4 10540					10489	17.15	0.48%
金条 3 400 29 4 10504					10393	11.23	1.06%
金3 400 31 4 10486					10395	8.33	0.87%
金3 400 34 4 10465					10408	8.56	0.54%
金3 400 37 4 10482					10415	8.50	0.64%
金3 400 41 4 10501					10426	10.03	0.71%
金3 400 45 4 10485					10405	9.66	0.76%
金3 400 51 4 10583					10538	10.70	0.43%
金3 400 58 4 10776					10751	12.38	0.23%
金3 400 67 4 10797					10785	15.36	0.11%
金3 400 81 4 10614					10627	22.75	−0.12%
金 4 480 33 4 13598					13567	19.24	0.23%
金 4 480 35 4 13643					13635	19.17	0.06%
金 4 480 37 4 13520					13498	16.29	0.16%
金 4 480 41 4 13460					13473	16.55	−0.10%
金 4 480 44 4 13568					13540	16.65	0.21%
金 4 480 49 4 13758					13772	17.88	−0.10%
金条 4 480 54 4 13760					13767	19.11	−0.05%
金 4 480 61 4 13791					13796	20.86	−0.04%
金 4 480 69 4 13966					13975	20.77	−0.06%
金 4 480 81 4 13975					14001	27.50	−0.19%
金 4 480 97 4 13775					13833	36.26	−0.42%

表11
黄金实例 5-8 的结果。

实例				公共数据	混合粒子群算法		
团体	n	C	原布_		解决方案	CPU 时间	改进 %
黄金5号	200 14 4 200 15 3				7622	7462	3.08
黄金5号	200 16 3 200 17 3				7424	7424	2.94
黄金5号	200 19 4 200 21 4				7491	7491	2.92
黄金5号	200 23 4 200 26 4				7434	7434	2.83
黄金5号	200 29 4 200 34 4				7576	7484	2.11
黄金5号	200 41 4				7596	7489	1.98
黄金5号					7643	7532	2.02
黄金5号					7560	7436	2.15
黄金5号					7410	7299	2.28
黄金5号					7429	7321	2.52
黄金5号					7241	7130	2.69
金6号	280 19 3 280 21 3				8624	8624	8.87
金6号	280 22 3 280 24 4				8628	8633	7.97
金6号	280 26 4 280 29 4				8646	8655	6.14
金6号	280 32 4 280 36 4				8853	8728	5.46
金6号	280 41 4 280 47 4				8910	8777	5.57
金6号	280 57 4				8936	8846	4.51
金6号					8891	8799	4.37
金6号					8969	8862	4.79
金6号					9028	8920	5.30
金6号					8923	8823	6.08
金6号					9028	8948	7.77
金7号	360	25	3		9904	9978	12.34
金7号	360	26	3		9888	9946	10.85
金7号	360	28	3		9917	9963	10.67
金7号	360	31	4		10021	9989	10.00
金7号	360	33	4		10029	9937	9.42
金7号	360	37	4		10131	10034	9.93
金7号	360	41	4		10052	9975	10.57
金7号	360	46	4		10080	10010	9.70
金7号	360	52	4		10095	10010	10.15
金7号	360	61	4		10096	10061	12.83
金7号	360	73	4		10014	9985	17.67
金 8 440 30 4					10866	10797	13.57
金 8 440 32 4					10831	10744	13.48
金 8 440 34 4					10847	10787	13.54
金 8 440 37 4					10859	10792	13.09
金 8 440 41 4					10934	10898	13.50
金 8 440 45 4					10960	10947	13.65
金 8 440 49 4					11042	11045	11.84
金 8 440 56 4					11194	11224	13.35
金 8 440 63 4					11252	11279	15.74
金 8 440 74 4					11321	11314	21.45
金 8 440 89 4					11209	11256	30.78

表12
黄金实例 9-12 的结果。

实例				公共数据	混合粒子群算法		
团体	n	厘米UB			解决方案	CPU 时间	改进 %
金9号	255	18	4		300 296 299 295	3.17	1.33%
金9号	255	19	4		296 293 290 289	3.05	1.34%
金9号	255	20	4		290 289 288 285	2.98	1.01%
金9号	255	22	4		292 291 297 293	2.91	0.34%
金9号	255	24	4		294 290 295 292	2.85	0.34%
金9号	255	26	4		296 294	2.76	1.04%
金9号	255	29	4			2.78	0.34%
金9号	255	32	4			3.03	1.35%
金9号	255	37	4			3.43	1.36%
金9号	255	43	4			4.09	1.02%
金9号	255	52	4			5.52	0.68%
黄金10号	323	22	4		367 373 361	5.60	−1.63%
黄金10号	323	24	4		365 359 361	5.28	−1.11%
黄金10号	323	25	4		361 365 367	5.20	−0.56%
黄金10号	323	27	4		370 373 379	5.27	−1.11%
黄金10号	323	30	4		385 389 400	5.43	−0.82%
黄金10号	323	33	4		402 398 399	5.40	−1.61%
黄金10号	323	36	4		393 395 387	5.61	−1.04%
黄金10号	323	41	4		389	6.18	−0.50%
黄金10号	323	47	4			7.10	−0.25%
黄金10号	323	54	4			8.77	−0.51%
黄金10号	323	65	4			12.11	−0.52%
金11号	399	27	5		457 452 455 456	8.42	1.09%
金11号	399	29	5		455 457 455 456	8.40	−0.22%
金11号	399	31	5		459 461 461 462	8.44	−0.44%
金11号	399	34	5		462 461 458 456	8.50	−0.22%
金11号	399	37	5		456 458 454 458	8.70	−0.44%
金11号	399	40	5		451 454	9.07	−0.22%
金11号	399	45	5			9.76	0.22%
金11号	399	50	5			10.98	0.44%
金11号	399	58	5			13.55	−0.44%
金11号	399	67	5			17.44	−0.88%
金11号	399	80	5			25.36	−0.67%
黄金12号	483	33	5		535 541 537	11.56	−1.12%
黄金12号	483	35	5		542 535 542	11.55	−0.93%
黄金12号	483	38	5		537 541 535	11.64	−1.31%
黄金12号	483	41	5		545 533 540	11.59	−0.74%
黄金12号	483	44	5		535 545 538	11.93	−1.87%
黄金12号	483	49	5		542 546 539	13.69	−1.31%
黄金12号	483	54	5		546 545 560	15.33	−1.87%
黄金12号	483	61	5		551	18.42	−0.74%
黄金12号	483	70	5			23.37	1.28%
黄金12号	483	81	5			30.51	0.18%
黄金12号	483	97	5			44.42	1.61%

表13
黄金实例 13-16 的结果。

实例			公共数据		混合粒子群算法		
团体	n	C	原布		解决方案 CPU t (s)	改进 %	
金色 13 252 17			4	552	549	2.85	0.54%
金色 13 252 19			4	549	544	2.67	0.91%
金色 13 252 20			4	548	540	2.70	1.46%
金色 13 252 22			4	548	540	2.65	1.46%
金色 13 252 23			4	548	540	2.65	1.46%
金色 13 252 26			4	542	535	2.68	1.29%
金色 13 252 29			4	540	534	2.77	1.11%
金色 13 252 32			4	543	538	3.01	0.92%
金色 13 252 37			4	545	543	3.41	0.37%
金色 13 252 43			4	553	549	4.08	0.72%
金色 13 252 51			4	560	554	5.17	1.07%
金色 14 320 22			4	692	690	4.87	0.29%
金色 14 320 23			4	688	685	4.64	0.44%
金色 14 320 25			4	678	676	4.41	0.29%
金色 14 320 27			4	676	676	4.38	0.00%
金色 14 320 30			4	678	680	4.42	−0.29%
金色 14 320 33			4	682	681	4.45	0.15%
金色 14 320 36			4	687	685	4.58	0.29%
金色 14 320 41			4	690	688	5.04	0.29%
金色 14 320 46			4	694	691	5.84	0.43%
金色 14 320 54			4	699	697	7.46	0.29%
金色 14 320 65			4	703	697	10.13	0.85%
金色 15 396 27			4	842	854	6.86	−1.43%
金色 15 396 29			4	843	852	6.89	−1.07%
金色 15 396 31			4	837	851	6.69	−1.67%
金色 15 396 34			4	838	852	6.85	−1.67%
金色 15 396 37			4	845	857	6.98	−1.42%
金色 15 396 40			4	849	856	7.40	−0.82%
金色 15 396 45			5	853	852	7.36	0.12%
金色 15 396 50			5	851	853	10.27	−0.24%
金色 15 396 57			5	850	853	12.35	−0.35%
金色 15 396 67			5	855	857	16.69	−0.23%
金色 15 396 80			5	857	858	24.89	−0.12%
金色 16 480 33			5	1030 1029 1028	11.45	0.10%	
金色 16 480 35			5	1026 1028 1028	10.97	0.19%	
金色 16 480 37			5	1032 1035 1028	10.83	−0.00%	
金色 16 480 41			5	1031 1031 1034	11.50	−0.29%	
金色 16 480 44			5	1022 1027 1013	11.93	−0.29%	
金色 16 480 49			5	1022 1012 1017	12.51	−0.29%	
金色 16 480 54			5	1018 1024 1018	14.74	−0.49%	
金色 16 480 61			5	1025	17.35	−0.89%	
金色 16 480 69			5		21.60	−0.49%	
金色 16 480 81			5		30.20	−0.59%	
金色 16 480 97			5		40.92	−0.69%	

表14
黄金实例 17-20 的结果。

实例	公司数据		混合粒子群算法		
	n	厘米UB	解决方案 CPU t (s) 420	改进 %	
金色 17 240 17	3	418	422	3.17	−0.49%
金色 17 240 18	3	419	422	3.07	−0.72%
金色 17 240 19	3	422	426	2.94	−0.00%
金色 17 240 21	3	425	426	2.89	−0.24%
金色 17 240 22	3	424	419	2.92	−0.47%
金色 17 240 25	3	418	415	2.68	−0.24%
金色 17 240 27	3	414	411	2.67	−0.24%
金色 17 240 31	4	421	406	2.77	2.38%
金色 17 240 35	4	417	403	2.97	2.64%
金色 17 240 41	4	412	404	3.61	2.18%
金色 17 240 49	4	414	413	4.13	2.42%
金色 18 300 21 4		592	587	5.01	0.84%
金色 18 300 22 4		594	590	4.98	0.67%
金色 18 300 24 4		592	587	4.05	0.84%
金色 18 300 26 4		590	580	4.17	1.69%
金色 18 300 28 4		577	569	4.04	1.39%
金色 18 300 31 4		578	572	3.63	1.04%
金色 18 300 34 4		582	574	3.69	1.37%
金色 18 300 38 4		586	580	4.34	1.02%
金色 18 300 43 4		594	584	4.70	1.68%
金色 18 300 51 4		601	591	5.81	1.66%
金色 18 300 61 4		599	588	8.09	1.84%
金色 19 360 25	10 925 10	807	807	54.79	12.76%
金色 19 360 26	924 4 808 4	807	807	52.66	12.60%
金色 19 360 28	811 4 797 5	813	813	9.26	−0.62%
金色 19 360 31	799 5 789 5	815	815	7.84	−0.49%
金色 19 360 33	788 5 800 5	802	802	7.07	−0.63%
金色 19 360 37	807 5 810	790	790	7.00	1.13%
金色 19 360 41		776	776	7.30	1.65%
金色 19 360 46		775	775	8.27	1.65%
金色 19 360 52		788	788	9.68	1.50%
金色 19 360 61		798	798	12.27	1.12%
金色 19 360 73		801	801	17.74	1.11%
金色 20 420 29	11 1220 1081 12 1232 1072	99.2	99.2	11.39%	
金色 20 420 31	12 1208 1060 1059 1056 5	84.19	84.19	12.99%	
金色 20 420 33	5 1052 1047 5 1052 1048 5	78.68	78.68	12.25%	
金色 20 420 36	1053 1052 5 1058	10.93	10.93	0.28%	
金色 20 420 39	1053 5 1058 1055 5 1049	10.14	10.14	0.48%	
金色 20 420 43	1054 5 1049 1045	9.58	9.58	0.38%	
金色 20 420 47		17.35	17.35	0.09%	
金色 20 420 53		18.52	18.52	0.47%	
金色 20 420 61		23.47	23.47	0.28%	
金色 20 420 71		30.54	30.54	0.47%	
金色 20 420 85		38.64	38.64	0.38%	

算法在 CluVRP 中找到的基准实例上进了测试
文学。该研究的主要贡献包括设计
一种新的混合 PSO 元启发式算法来求解 CluVRP
并为总共 138 个找到新的最知名的解决方案
293 个具有平均 CPU 性能的基准测试实例中的实例
时间6.99秒。它还通过添加新的内容对本研究做出了贡献
PSO 算法的特点,例如使用两种类型
个人最佳解决方案的粒子和改进方案。
在改进方案中,个人最佳解决方案
通过采用扰动和
VNS法。因此,所提出的算法具有巨大的潜力
用于解决 VRP 其他变体的实例。凭借在相对可接受的 CPU 时间上提供
高质量解决方案的能力,
算法具有在许多实际场景中使用的视角
例如具有二氧化碳碳排放上限的配送物流
罚款、易腐烂物品问题和运输
军事动等方面的问题。像所有研究工作一样,这
工作也存在一些局限性和未来的研究方向。
VRP 的许多属性,例如时间窗、碳排放、回程和多仓库,都可以通过 CluVRP 添加到

捕捉真实世界的场景。虽然所提出的算法是
旨在单独解决 CluVRP,它可以轻松扩展以解决
VRP 的其他变体。未来的研究工作还可以探索
将 PSO 与其他元启发法相结合的可能性,例如
遗传算法、禁忌搜索、模拟退火等

CRedit 作者贡献声明

Md. Anisul Islam:分析问题,设计
方法论,分析和解释数据,概念化
解决方案技术,进实验,编写

纸。Yuvraj Gajpal:设计方法、分析和
解释数据,概念化解决方案技术,审查和编辑论文。 Tarek Y. ElMekkawy:设计了

方法论,分析和解释数据,概念化
解决方案技术,审阅并编辑了论文。

竞争利益声明

作者声明,他们没有已知的可能出现的相互竞争的经济利益或个人关系

影响本文报告的工作。

致谢

作者感谢曼尼托巴省政府通过教育部提供的资金

高级学习,以及曼尼托巴大学的支持
这项研究。作者感谢匿名审稿人的贡献
提出宝贵意见。

附录.详细计算结果

参见表8-14。

参考

[1] C. Exposito-Left,A. Rossi,M. Sevaux,两级解决方案
为了解决集群化的车辆路径问题,Comput.工业。
工程师。 91 (2016)274-289。
[2] P. Toth, D. Vigo,模型、松弛和精确方法
容量车辆路径问题,离散应用。数学。 123 (2002)
487-512。
[3] T. Barthélemy,A. Rossi,M. Sevaux,K. Sörensen,元启发式方法
集群 VRP,见:EU/ME 2010元启发式 10 周年
社区,洛里昂,法国,2010 年。
[4] PC Pop,J. Kara,AH Marc,广义的新数学模型
车辆路径问题及其扩展,应用。数学。模型。 36 (2012)
97-107。
[5] M. Battarra,G. Erdo an,D. Vigo,集群车辆的精确算法
路由问题,操作员。资源。 62 (2014)58-71。
[6] AH Marc,L. Fuksz,PC Pop,D. Danciulescu,一种新颖的混合算法
用于解决集群车辆路径问题,见:E. Onieva,J. Santos,
E. Osaba,E. Quintián (主编),混合人工智能系统,Springer,
2015 年,第 679-689 页。
[7] T. Vidal,M. Battarra,A. Subramanian,G. Erdogan,混合元启发法
对于集群车辆路径问题,Comput.歌剧。资源。 48 (2015)
87-99。
[8] C. Defryn,K. Sörensen,快速两级变量邻域搜索
集群车辆路径问题,Comput.歌剧。资源。 83 (2017)78-94。
[9] PC Pop,L. Fuksz,AH Marc,C. Sabo,一种新颖的两级优化
集群车辆路径问题的方法,Comput.工业工程115
(2018)304-318。
[10] T. Hintsch,S. Irnich,聚类的大型多邻域搜索
车辆路径问题,欧洲 J. Oper。资源。 270 (2018)118-131。
[11] T. Hintsch,软聚类的大型多邻域搜索
车辆路线问题,计算。歌剧。资源。 129(2021)105132。
[12] M. Sevaux,K. Sörensen,大型集群路由问题中的哈密顿路径,见:2008 年 EU/Meeting 元启发学研
讨会论文集
物流和车辆路线,欧盟/中东,卷。 8,2008 年,第 411-417 页。
[13] V. Schmid,KF Doerner,G. Laporte,供应中出现的丰富路由问题
连锁管理,欧洲 J. Oper。资源。 224 (2013)435-448。
[14] A. Subramanian,车辆路径问题的启发式、精确和混合方法 (博士论文),尼泰罗伊联邦弗鲁米嫩塞
大学 (巴西,
2012年。
[15] RC Eberhart,J. Kennedy,使用粒子群理论的新优化器,
见:第六届国际微型机械研讨会论文集
和人类科学,1995 年,第 39-43 页。
[16] TJ Ai,V. Kachitvichyanukul,车辆的粒子群优化
同时取货和送货的路由问题,计算。歌剧。
资源。 36 (2009)1693-1702。
[17] Y. Marinakis,GR Iordanidou,M. Marinaki,粒子群优化
具有随机需求的车辆路径问题,应用。软计算。
13 (2013)1693-1704。
[18] N. Norouzi,M. Sadegh-Amalnick,M. Alinaghiyan,粒子评估
周期性车辆路径问题中的群体优化,测量
62 (2015)162-169。

[19] DC Hop,N. Van Hop,TTM Anh,集成码头起重机和堆场卡车调度问题的自适应粒子群优化,Comput.工业. 153 (2021)107075。

[20] IH Dridi,EB Alaïa,P. Borne,H.Bouchriha,使用 PSO 算法优化时间窗和多车辆的多仓库取货和送货问题,Int. J.产品.资源。 58 (14) (2020)1-14。

[21]晁问,刘世,钱强,谭志,王红,需求不确定下的中欧运输网络优化,亚太地区。 J. 歌剧.资源。(2021)。

[22] SN Sahu,Y. Gajpal,S. Debbarma,基于两个代理的单机调度,具有切换时间以最小化总加权完成时间和完工时间目标,Ann.歌剧.资源。 269 (2018)623-640。

[23] J. Li,Y. Gajpal,AK Bhardwaj,H. Chen,Y. Liu,最大化净收入的两代理单机订单接受调度问题,Complexity (2021) <http://dx.doi.org/10.1155/2021/6627081>。

[24] N. Mladenovic,P. Hansen,可变邻域搜索,计算.歌剧.资源。 24 (11) (1997)1097-1100。

[25] MA Islam,Y. Gajpal,碳排放上限下传统和绿色车辆成分的优化,可持续发展 13 (12) (2021) 6940, <http://dx.doi.org/10.3390/su13126940>。

[26] P. Hansen, N. Mladenovic,可变邻域搜索,见:F. Glover, G. Kochenberge (编),元启发式手册,Boston,Kluwer,2003 年,第145-184 页。

[27] Y. Marinakis,M. Marinaki,G. Dounias,车辆路径问题的混合粒子群优化算法,Eng.应用.阿蒂夫.英特尔。 23 (2010)463-472。

[28] FP Goksal,I. Karaoglan,F. Altiparmak,针对同时取货和送货的车辆路径问题的混合离散粒子群优化,Comput.工业工程65 (2013)39-53。

[29] Marinakis Y.,Marinaki M.,Migdalas A.,带时间窗的车辆路径问题的多自适应粒子群优化,Info. 科学。 8 (10) (2019)2583-2589。

[30]邹金. Li,X. Li,一种带时间窗的多目标取货和配送问题的混合粒子群优化算法,J. 计算。 8 (10) (2013)2583-2589。

[31] S. 张,M. 陈,W. 张,具有随机需求的电动汽车运输中的新型位置路由问题,J. Cleaner Prod. 221 (2019) 567-581。

[32] P. Hansen,N. Mladenovic,可变邻域搜索:原理和应用,European J. Oper.资源。 130 (3) (2001)449-467。

[33] BF Moghaddam,R. Ruiz,SJ Sadjadi,需求不确定的车辆路径问题:先进的粒子群算法,Comput.工业. 62 (2012)306-317。

[34] G. Moslehi,M. Mahnam,使用粒子群优化和局部搜索解决多目标灵活作业车间调度问题的帕累托方法,Int. J.产品.经济。 129 (1) (2011)14-22。

[35] H. Liu,A. Abraham,O. Choi,SH Moon,多目标灵活作业车间调度问题的可变邻域粒子群优化,计算讲座笔记.科学。 4247 (2006)197-204。

[36] P. Pongchairerks,V. Kachitvichyanuku,针对作业车间调度问题的带 PSO 的 VNS 算法和不带 PSO 的 VNS算法的比较,Int. J. 计算机.科学。 1 (2) (2007)179-191。

[37] AF Ali,AE Hassanien,V. Snasel,MF Tolba,一种新的混合粒子群优化与可变邻域搜索,用于解决无约束全局优化问题,见:P. Kromer 等人。(编),第五届实习生论文集.会议.关于创新。 151 仿生计算.和应用。 IBICA 2014,载于:智能系统和计算的进展,303,瑞士施普林格国际出版公司,2014 年, http://dx.doi.org/10.1007/978-3-319-08156-4_16c。

[38] L.Zhang,J.Wu,一种基于 PSO 的混合元启发式置换流商店调度问题,科学.世界杂志 (2014)1-8。

[39] BF Gumaida,J. Luo,用于无线传感器网络中定位增强的可变邻域搜索的混合粒子群优化,Appl.英特尔。 49 (2019)3539-3557。

[40] Y. Marinakis,A. Migdalas,A. Sifaleras,用于约束最短路径问题的混合粒子群优化-变量邻域搜索算法,European J. Oper.资源。 261 (2017)819-834。

[41]蔡L.,吕文,肖L.,徐Z.,带速度变量的互联自动车辆路径问题中的总碳排放最小化,专家系统.应用。 165 (2021)113910。

[42] M. Ranjbar,RG Saber,单站多产品转运调度的变量邻域搜索算法,Appl.软计算。 98 (2021) 106736。

[43] MA Islam,Y. Gajpal,TY ElMekkawy,碳排放上限下基于混合车队的绿色集群物流问题,可持续城市协会。 72 (2021)103074。

[44] A. Subramanian,LMA Drummond,C. Bentes,LS Ochi,R. Farias,同时取货和送货的车辆路径问题的并启发式,Comput.歌剧.资源。 37 (11) (2010)1899-1911。

[45] T. Bektas,G. Erdogan,S. Ropke,广义车辆路径问题的公式和分支割算法,Transp.科学。 45 (2011) 299-316。

[46] AE Ezugwu,AO Adewumi,ME Frıncu,基于模拟退火的旅商问题共生生物搜索优化算法,专家系统.应用。 77 (2017)189-210。

[47] G. Macrina,LDP Pugliese,F. Guerriero,G. Laporte,具有部分电池充电和时间窗口的绿色混合车队车辆路线,Comput.歌剧.第 101 号决议 (2019) 183-199。

阿尼苏尔·伊斯兰博士(Md. Anisul Islam)是一名博士。加拿大温尼伯曼尼托巴大学机械工程系学生。他获得了硕士学位。工业工程。他的研究兴趣是绿色车辆路径问题、运输、配送和物流问题的优化技术。

Yuvraj Gajpal是加拿大曼尼托巴大学温尼伯分校阿斯珀商学院供应链管理助理教授。他拥有博士学位。加拿大汉密尔顿麦克马斯特大学德格魯特商学院管理科学学士学位和印度马德拉斯印度理工学院(IIT)工业管理硕士学位。他的研究兴趣主要在于启发式和元启发式在运输和物流管理中的应用。他曾在《Computers and Operations Research》、《European Journal of Operations Research》、《International Journal of Production Economics》、《Annals of Operations Research》、《Reliability Engineering and Systems Safety》、《Construction Management and Economics》和《Journal of the Operational Research Society》等领先研究期刊上发表论文。他是许多国际期刊的审稿人,例如Computers and Operations Research,European Journal of Operations Research、Computers and Industrial Engineering,Journal of Heuristics和Transportation Research Part E。

TY ElMekkawy 博士是卡塔尔大学教授。他的研究重点之一与制造系统的调度和性能优化有关。他的研究得到了加拿大自然科学与工程研究委员会 (NSERC)的支持。他获得了加拿大工业和医疗保健组织的研究资助,例如 Motor Coach (曼尼托巴省)、温尼伯地区卫生局 (WRHA)、曼尼托巴患者访问网络 (MPAN)、曼尼托巴健康局和 CancerCare Manitoba。他的论文发表在《Journal of Renewable Energy》、《International Journal for Energy Research》、ASME Transactions,AMJ,IJPR,IJPE,IJCIM、IJAMT,IJOR 和 EJE 等高影响力期刊上。