DIFF - DIstribuição de Fármacos a Farmácias*

João Miguel Monteiro e Jorge David Pacheco

Faculdade de Engenharia da Universidade do Porto, Dr. Roberto Frias, 4200-465
Porto, Portugal
http://www.fe.up.pt
{up201705580,up201705754}@fe.up.pt

Resumo. Este trabalho, realizado no âmbito da Unidade Curricular de Programação em Lógica, tem por objetivo otimizar um problema utilizando restrições. Este problema consiste num conjunto de farmácias a ser visitadas numa determinada ordem e com um número de carrinhas disponíveis. O objetivo é minimizar a distância percorrida pelas carrinhas, assim como o número das mesmas.

Palavras-chave: Otimização · Ciência de computadores · Eficiência.

1 Introdução

Este projeto foi desenvolvido no âmbito da unidade curricular de Programação Lógica de 3º ano do curso Mestrado Integrado em Engenharia Informática e Computação, usando o Sistema de Desenvolvimento SICStus Prolog, tendo como tema um problema de otimização com restrições. O grupo escolheu o problema de distribuição de fármacos que consiste em encontrar um equilíbrio entre o número de carrinhas a usar e entre a distância percorrida por cada uma.

2 Descrição do Problema

A empresa DIstribuição de Fármacos a Farmácias (DIFF) necessita de programar o seu horário de distribuição de forma a levar todos os medicamentos necessários a todas as farmácias por eles servidas.

A distribuidora opera das 10h às 22h, sendo que cada farmácia tem um horário específico em que está disponível para receber as encomendas. Os requisitos de cada farmácia podem ser medidos em volume da encomenda, sendo que as carrinhas de entrega têm um limite de volume de transporte. Cada farmácia tem uma localização, assim como a empresa de distribuição. A entrega da mercadoria tem uma duração de 30 minutos, e o tempo de viagem entre cada ponto (farmácias e sede) é conhecido. Pretende-se minimizar o número de carrinhas de entrega necessárias, assim como a distância total percorrida pelas carrinhas. Modele este problema como um problema de otimização / satisfação de restrições e resolva-o em PLR de forma a ser possível satisfazer problemas desta classe com

* FEUP-PLOG, TURMA 3MIEIC2, GRUPO DIFF3.

diferentes parâmetros (número de carrinhas disponíveis, número de farmácias a servir, volumes das encomendas e capacidade das carrinhas, distâncias entre locais, etc.).

3 Abordagem

Para resolver este problema de otimização, foi criado um predicado de farmácias (farmacia/5), com os seguintes campos:

```
farmacia(+Name, +Id, +Volume_Encomenda, +HoraInicio, +HoraFim).
```

Fig. 1. Predicado farmacia/5

Utilizamos uma matriz para representar a distância, à qual chamamos distTime.

```
distTime([
    0, 20, 15, 30, 47, 28,
   20,  0,  2, 32, 23, 43,
   15,  2,  0,  3, 35, 26,
   30, 32,  3,  0, 21, 60,
   47, 23, 35, 21,  0, 42,
   28, 43, 26, 60, 42,  0
]).
```

Fig. 2. distTime/1.

Para facilitar a resolução do nosso problema, todos os dados relacionados com o tempo são apresentados e calculados em minutos. A empresa em questão só opera entre as 10 e as 22 horas como indicado na descrição do problema, tendo portanto um intervalo de 720 minutos para completar as entregas (sendo que cada tem um intervalo de 30 minutos associado). Também consideramos que a uma unidade de distância corresponde 1 minuto.

3.1 Variáveis de decisão

De modo a calcular uma solução desejada, foi criada uma variável D, onde é armazenada a informação relativa à distância mínima encontrada. Paralelamente, existe também uma variável NVans, que representa o número de carrinhas a serem utilizadas.

3.2 Restrições

Foi necessário criar uma restrição que se certifica que o tempo despendido por uma determinada carrinha não supera os 720 minutos, *i.e.*, o tempo total disponível por dia.

3.3 Função de avaliação

De modo a encontrar uma solução otimizada para o nosso problema, pensamos em 3 métodos diferentes. Dois destes métodos usam a função *cumulatives*, em que cada *machine* representa uma das carrinhas a ser usadas, enquanto que o terceiro não.

Embora distintos, todos os métodos partem da mesma base de guardar uma lista com a informação de qual carrinha visitou qual farmácia.

Em todos os casos, a melhor solução implica a menor distância percorrida por todas as carrinhas. Para tal, são guardados os tempos que indicam o fim do percurso de cada carrinha, uma vez que, quanto menor for o tempo do percurso, menor é a distância percorrida.

3.4 Estratégia de pesquisa

Na etiquetagem do nosso *labeling*, optamos por utilizar o método de minimizar um valor, sendo este a soma total dos tempos gastos (ou distâncias percorridas) por cada carrinha. Com este método fomos capazes de chegar a uma solução ideal do nosso problema.

4 Visualização da solução

Toda a informação visualizada é gerida no predicado *diff*. Este predicado tem como objetivo mostrar um menu inicial, onde o utilizador poderá escolher entre várias opções, desde o tamanho do problema ao número de carrinhas disponíveis. Após todos os cálculos e otimizações realizadas, o resultado é revelado na consola por este mesmo predicado.

5 Resultados

Para resolver este problema, foram tidos em conta 3 dimensões diferentes: 5, 10 e 15 farmácias.

Inicialmente, vamos ter em conta a resolução deste problema com 1 carrinha.

Dimensão do Problema	Distância Calculada	Tempo (ms)
5	238	14
10	407	153
15	566	43184

Quando se começa a usar valores superiores ($nCarrinhas > 1$) para o número de carrinhas o tempo de solução do problema começa a aumentar exponencialmente, uma vez que o número de possíveis soluções aumenta, também, da mesma forma:

$$nFarmacias^{nCarrinhas}$$

6 Conclusões e Trabalho Futuro

Ao longo deste projeto foram encontradas vários obstáculos na resolução do problema, nomeadamente na escolha do método de otimização, alguns dos quais foram ultrapassados após uma analise cuidada dos slides fornecidos e da ajuda do professor.

Como tal, é de notar que existem aspetos que poderiam ser melhorados, como a escolha de um método mais eficiente e otimizado, dado que a nossa solução encontra-se limitada dado o tempo que o programa demora a resolver o problema dependendo da sua dimensão.

References

1. SWI-Prolog Homepage, https://www.swi-prolog.org/. Last accessed 5 Jan 2020
2. The GNU Prolog web site, http://www.gprolog.org/. Last accessed 5 Jan 2020
3. SICStus Prolog, https://sicstus.sics.se/. Last accessed 5 Jan 2020