

Greedy Randomized Adaptive Search Procedure com Path-Relinking para o vértice p- Problema do centro

Ai-Hua Yin¹, Membro, CFF, Tao-Qing Zhou^{2,3}, Jun-Wen Ding^{2,*}, Qing-Jie Zhao², e Zhi-Peng Lv²

¹ Escola de Engenharia de Software e Comunicação, Universidade de Finanças e Economia de Jiangxi

Nanchang 330013, China

² Escola de Ciência e Tecnologia da Computação, Universidade de Ciência e Tecnologia de Huazhong, Wuhan 430074, China

³ Departamento de Ciência da Computação, Escola de Engenharia da Informação, Universidade de Zhejiang Agricultura e Florestal Hangzhou 311300, China

E-mail: aihuayin@jxufe.edu.cn ; qqis@163.com ; dingjunwen@hust.edu.cn ; qingjie.zqj@alibaba-inc.com
zhipeng.lui@gmail.com

Recebido em 29 de agosto de 2016; revisado em 28 de março de 2017.

Abstrato O p- O problema do centro consiste em escolher um subconjunto de vértices em um grafo não direcionado como facilidades, a fim de minimizar a distância máxima entre um cliente e sua instalação mais próxima. Este artigo apresenta um procedimento de busca adaptativa aleatória gananciosa com algoritmo de reconexão de caminho (GRASP / PR) para o p- problema central, que combina GRASP e reconexão de caminho. Cada iteração do GRASP / PR consiste na construção de uma solução gulosa aleatória, seguida de um procedimento de busca tabu. A solução resultante é combinada com uma das soluções de elite por path-relinking, que consiste em explorar trajetórias que conectam soluções de alta qualidade. Experimentos mostram que GRASP / PR é competitivo com os algoritmos de última geração na literatura em termos de qualidade de solução e eficiência computacional. Especificamente, ele melhora virtualmente os resultados anteriores mais conhecidos para 10 entre 40 instâncias grandes, ao mesmo tempo em que combina os resultados mais conhecidos para outras.

Palavras-chave p- problema no centro, pesquisa tabu, reconexão de caminho, localização da instalação

1. Introdução

O objetivo do p- o problema central é localizar p instalações e atribuir clientes a eles de modo a minimizar a distância máxima entre um cliente e sua instalação mais próxima. Cada cliente é atendido apenas por uma instalação. Na forma matemática, a definição do p- O problema do centro pode ser descrito como segue.

Deixar $G = (V, E)$ seja um gráfico não direcionado com vértice definir $V = \{v_1, \dots, v_n\}$ e conjunto de borda E . A distância ser- entre quaisquer dois vértices v_{eu} e v_j é dado por d_{v_{eu}, v_j} . O p- problema central é encontrar $S \subseteq V$ ($|S| = p$) de modo que a função objetivo:

$$f(S) = \max \{ \min_{v_{eu}, v_j} d_{v_{eu}, v_j} \},$$

Onde $v_{eu} \in S$, $v_j \in V - S$, é minimizado.

O p- problema central provou ser um problema NP-difícil [1-2] e um grande número de métodos para resolver este problema foi relatado na literatura. Entre eles estão várias abordagens exatas: Hakimi [3]

primeiro definiu o p- problema central e propôs vários métodos baseados em gráficos para resolvê-lo. Kariv e Hakimi [1-2] projetou um $O(|E| p n^{2p-1} / (p-1)! \lg n)$ e um $O(|E| p n^{2p-1} / (p-1)!)$ ("| |" Significa o número de elementos no conjunto) algoritmo para encontrar um algoritmo absoluto p- centro em uma rede ponderada por vértice e uma rede não ponderada por vértice, respectivamente, e estendeu esses resultados, dando um Sobre $\lg_{p-2} n$ algoritmo para encontrar um absoluto

p- centro (onde $3 \leq p < n$) e um Sobre $\lg_{p-1} n$ algoritmo para encontrar um vértice p- centro (onde $2 \leq p < n$).

Minieka [4] introduziu um algoritmo para determinar a distância mínima de limiar onde todos os clientes estão cobertos

Regular Paper

The research was supported by the National Natural Science Foundation of China under Grant Nos. 61370183 and 61262011.

*Corresponding Author

©2017 Springer Science + Business Media, LLC & Science Press, China

ered por instalações resolvendo uma série finita de conjuntos mínimos cobrindo problemas. Com base neste método, Daskin [5] apresentou a formulação matemática do p- problema central e desenvolveu um método de bissecção para reduzir a lacuna entre os limites superior e inferior das soluções ótimas. Mais tarde, Daskin [6] melhorou seu algoritmo substituindo o problema da cobertura do conjunto pelo subproblema da cobertura do conjunto máximo e o resolveu com a relaxação de Lagrange. Ilhan e Pinar [7] propôs um método de duas fases para este problema, onde primeiro resolve os problemas de viabilidade para encontrar limites inferiores com base na formulação de programação linear e, em seguida, utiliza problemas de viabilidade para verificar soluções viáveis. Eles resolveram as 84 instâncias da biblioteca ORLIB com até 900 vértices para a otimização pela primeira vez. Elloumi et al. [8] apresentou uma formulação de programação linear inteira para o p- centro do problema com um número polinomial de variáveis e restrições, e propôs um algoritmo de busca binária para resolver instâncias com até 1 817 vértices pela primeira vez. Recentemente, Al-Khedhairi e Salhi [9] modificaram as propostas (IP* e Daskin*) para os algoritmos propostos em [5, 7] para melhorar sua eficiência. Especificamente, para o algoritmo proposto em [5], o autor usou limites inferiores e superiores iniciais mais estreitos e um método de busca binária mais apropriado para reduzir o número de subproblemas. Para o algoritmo proposto em [7], os autores reduziram o número de iterações ILP necessárias para encontrar a solução ótima. IP* e Daskin* foram testados nos conjuntos de benchmarks bem conhecidos OR-Library e TSPLIB com resultados encorajadores.

Além das conquistas maravilhosas na área de métodos exatos para o p- problema central, uma variedade de algoritmos heurísticos e metaheurísticos foram propostos para resolvê-lo. Hochbaum et al. [10] introduziu um algoritmo de 2 aproximações para este problema. Martinich [11] propôs uma abordagem de fechamento de vértice para investigar este problema e provou que pode resolver de forma otimizada alguns casos especiais e obter soluções muito boas para problemas onde p é grande em relação ao número de vértices n. Plesnik [12] apresentou um algoritmo polinomial com uma razão de erro de pior caso de 2 para o p- problema do centro. Recentemente, Mladenović et al. [13] apresentou uma pesquisa de vizinhança de variável básica (VNS) e duas heurísticas de pesquisa tabu (TS) para o p- problema central sem desigualdade de triângulo. Experimentos em todas as instâncias da ORLibrary e partes das instâncias TSPLIB mostraram que os algoritmos VNS e dois TS propostos superam os métodos de pesquisa binários anteriores, e

VNS é medianamente melhor do que TS para instâncias com grandes p. Hassin et al. [14] introduziu uma estratégia de busca local para resolver o p- problema central com um objetivo mín-máx (ou máx-mín) onde as soluções são comparadas lexicograficamente ao invés de por suas piores coordenadas. Caruso et al. [15] apresentou Dominante, um algoritmo metaheurístico para resolver o p- problema do centro. Experimentos em todas as instâncias da biblioteca OR indicaram que o Dominante tem bom desempenho na maioria dos casos para encontrar soluções ótimas ou quase ótimas em um período de tempo razoável. Pacheco e Casado [16] propôs um algoritmo metaheurístico (SS) baseado na abordagem de scatter search para dois problemas de localização com poucas facilidades ($p \leq 10$) e aplicou-o para resolver instâncias de dados reais de recursos de saúde. Davidović et al. [17] propôs uma variante do algoritmo de otimização de colônias de abelhas (BCOI) com base no conceito de melhoria para o p- problema do centro. Os experimentos nas instâncias da biblioteca OR mostraram que o BCOI é competitivo em comparação com o VNS e o SS. Yurtkuran e Emel [18] propôs um algoritmo de colônia de abelhas artificial modificada (M-ABC) que se beneficia de uma variedade de estratégias de busca para equilibrar a intensificação e diversificação da busca e aplicou esquemas de codificação baseados em chave aleatória para resolver p- problema do centro. Experimentos nas instâncias da biblioteca OR revelaram a eficácia do M-ABC quando comparado com VNS, SS e BCOI. Além disso, alguns estudos teóricos da

p- o problema do centro é apresentado em [19-21]. Arostegui et al. [22] comparou o desempenho relativo de pesquisa tabu (TS), recozimento simulado (SA) e algoritmos genéticos (GA) em vários tipos de p- centralize o problema em condições de tempo limitado, solução limitada e irrestrita. Pullan [23] propôs um algoritmo genético memético (PBS) para o p- problema central, que aplica crossover de fenótipo e operadores de mutação direcionada para gerar novos pontos de partida para uma pesquisa local. O PBS foi considerado a melhor heurística por poder obter as soluções ótimas para todas as instâncias pequenas ($|V| < 1\,000$) da OR-Library e TSPLIB e soluções de alta qualidade para grandes instâncias ($|V| > 1\,000$) do TSPLIB quando comparado com as outras heurísticas mencionadas acima.

Este artigo apresenta um GRASP com algoritmo de reconexão de caminho (denotado por GRASP / PR) para resolver o p- problema do centro. GRASP / PR foi introduzido pela primeira vez em [24], que foi aplicado com sucesso a vários problemas combinatórios clássicos, incluindo o problema de atribuição de três índices [25], o problema de Steiner em gráficos [26],

① <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/pmedinfo.html>, Jan. 2017.

② <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/>, Jan. 2017.

o problema de agendamento da oficina [27], o problema de atribuição quadrática [28], o problema de corte máximo [29], etc. Cada iteração do algoritmo GRASP / PR busca obter uma solução ótima local, seguida por um pathrelinking que explora trajetórias entre a solução ótima local atual e as soluções ótimas anteriores preservadas no conjunto elite. Tanto no GRASP quanto no procedimento de revinculação de caminho, uma pesquisa tabu é adotada para intensificar a pesquisa. Realizamos o algoritmo GRASP / PR proposto em dois conjuntos de 124 instâncias de benchmark comumente usadas na literatura e descobrimos que GRASP / PR virtualmente melhora as melhores soluções anteriores para 12 instâncias e combina soluções anteriores mais conhecidas para as outras, demonstrando a eficiência do algoritmo proposto.

O restante deste trabalho está organizado da seguinte forma. A seção 2 descreve a estrutura geral e os detalhes do algoritmo GRASP / PR. A seção 3 relata os resultados computacionais e a comparação com algoritmos de última geração na literatura. A seção 4 analisa e discute alguns ingredientes essenciais do algoritmo GRASP / PR. Finalmente, a Seção 5 faz uma revisão da principal contribuição de nosso trabalho e delinea as futuras direções de pesquisa.

2 Algoritmo GRASP / PR

2.1 Estrutura Geral do Algoritmo GRASP / PR

GRASP é uma metaheurística multi-start para problemas combinatórios em que cada iteração consiste em dois

fases: construção e pesquisa local. A fase de construção constrói uma solução viável, onde a vizinhança definida é investigada até que um mínimo local seja encontrado durante a fase de busca local. Equilibrar intensificação e diversificação desempenha um papel essencial em quaisquer métodos heurísticos [30-33]. A reconexão de caminhos é uma abordagem para integrar a intensificação e a diversificação na pesquisa. Consiste em explorar trajetórias que conectam soluções de alta qualidade. A trajetória é gerada selecionando movimentos introduzidos nos atributos da solução inicial da solução guia. O objetivo do path-relinking é integrar características de boas soluções obtidas por cada iteração do GRASP, em novas soluções geradas em iterações subsequentes. No GRASP puro (ou seja, GRASP sem reconexão de caminhos), todas as iterações são independentes e, portanto, a maioria das boas soluções são simplesmente “esquecidas”. A reconexão de caminhos tenta mudar essa situação mantendo soluções anteriores e usando-as como “guias” para acelerar a convergência para uma solução de alta qualidade.

A arquitetura geral do GRASP / PR é descrita no Algoritmo 1. Por conveniência, adotamos as seguintes notações no algoritmo GRASP / PR:

- N_v : uma lista ordenada dos vértices adjacentes de v sequenciado por ordem não decrescente da distância até v ;
- N_{vk} : o primeiro k vértices em N_v . Onde k é o índice da instalação em N_v isso serve $v(N_v[0], \dots, N_v[k-1])$;
- $d(S_{eu}, S_j)$: a distância $d(S_{eu}, S_j)$ entre duas soluções S_{eu} e S_j , que é o tamanho da diferença simétrica entre as duas soluções S_{eu} e S_j dado

Algorithm 1 . Pseudo-Code of GRASP/PR Algorithm for p -Center

```

1: Input: problem instance
2: Output: the best found solution  $S^*$ 
3:  $P \leftarrow \emptyset$ 
4: while stopping condition is not satisfied do
5:    $S \leftarrow GreedyRandomized()$ 
6:    $S^t \leftarrow TabuSearch(S)$ 
7:   if  $P$  is full then
8:     Select an elite solution  $S^e \in P$  at random
9:      $S^r \leftarrow PathRelinking(S^t, S^e)$ 
10:    if  $S^r \notin P$  and  $f(S^r) \leq \max\{f(S) | S \in P\}$  then
11:      Let  $P' = \{S | S \in P, f(S) \geq f(S^r)\}$ 
12:      Let  $S^w \in P'$  be the most similar solution to  $S^r$ , i.e.,  $S^w = \operatorname{argmin}\{d(S, S^r) | S \in P'\}$ 
13:      If there is more than one  $S^w$ , randomly select one as  $S^w$ 
14:       $P \leftarrow P \cup \{S^r\}$ ,  $P \leftarrow P \setminus \{S^w\}$ 
15:    end if
16:  else
17:    if  $S^t \notin P$  then
18:       $P \leftarrow P \cup \{S^t\}$ 
19:    end if
20:  end if
21: end while
22:  $S^* \leftarrow \operatorname{argmin}\{f(S) | S \in P\}$ 

```

de:

$$d(S_{eu}, S_j) = p - |S_{eu} \cap S_j|. \quad (1)$$

No início, o conjunto de elite P está vazio, e as soluções ótimas locais obtidas por GRASP são adicionadas em P se eles são diferentes das soluções já em

P . Uma vez que o conjunto elite está completo, a reconexão de caminho é invocada após cada procedimento GRASP, onde uma solução elite $S_e \in P$ é selecionado aleatoriamente e combinado com a solução S obtido por GRASP por meio de reconexão de caminho.

Se a solução combinada S_r obtido por pathrelinking não está no conjunto de elite e seu valor objetivo não é maior que o da pior solução no conjunto de elite, então ele é inserido no conjunto de elite. Além disso, a solução que mais se assemelha a S_r (com a menor distância de acordo com (1)) e não melhor do que S_r

é excluído do conjunto elite. Observe que se houver mais de uma solução mais semelhante, que não é melhor do que S_r , GRASP seleciona um aleatoriamente e o exclui do conjunto elite. Este esquema mantém constante o tamanho do conjunto de elite e tenta manter a diversificação do conjunto. A condição de parada de GRASP / PR é que a pesquisa atinge um número pré-definido de iterações $eu_{máx}$. Nas subseções a seguir, os principais componentes do GRASP / PR (inicial ganancioso e aleatório procedimento de solução, pesquisa tabu e reconexão de caminho) são descritos em detalhes.

2.2 Solução Inicial

Na fase de construção do GRASP, uma solução inicial é construída iterativamente, um elemento de cada vez. Algoritmo 2 dá o pseudo-código do ganancioso e

procedimento construtivo randomizado para GRASP / PR. Primeiro, ele seleciona aleatoriamente um vértice $v \in V$ como a primeira facilidade e adiciona-o ao S . Obviamente, v é a facilidade mais próxima para todos os vértices em V . Em segundo lugar, ele seleciona o vértice cliente w ($w \in V$, $w \in S$) que tem o maior distância para suas instalações C_f . Em seguida, ele gera um número aleatório entre 0 e 1. Se este número for menor que α , ele seleciona aleatoriamente um vértice v da lista N_{sem} , e adiciona a S ; caso contrário, ele seleciona aleatoriamente um vértice v da lista N_c e adiciona a S . Parâmetro α controla a quantidade de ganância e aleatoriedade em o algoritmo. Um valor $\alpha = 1$ corresponde a um procedimento de construção ganancioso, enquanto $\alpha = 0$ produz construção aleatória. Este procedimento é repetido até o tamanho de S alcançar p .

2.3 Procedimento de pesquisa tabu

A busca tabu é um algoritmo de otimização inteligente inicialmente introduzido em [34] e tem sido aplicado a vários problemas de otimização combinatória [35-37]. Nosso O procedimento de pesquisa tabu é formado pela adição de uma estratégia tabu específica à pesquisa local apresentada em [23]. A vizinhança usada em [23] é definida como trocando um vértice em N_{sem} com uma facilidade na solução atual S . Deixar troca (i, j) ($i \in N_{sem}, j \in S$) denotam um movimento que troca um vértice eu dentro N_{sem} e uma instalação j dentro S . Para evitar andar de bicicleta, proibimos o movimento troca (i, j) ser realizado no próximo tt iterações (chamadas de posse tabu) quando uma mudança troca (i, j) é desempenhado. As informações para proibição de movimentação são mantidas na lista tabu TL onde o valor de TL (i, j) é o número da iteração quando

Algorithm 2. Pseudo-Code of the Greedy and Randomized Constructive Procedure

```

1: Input: problem instance, parameter  $\alpha$ 
2: Output: an initial solution  $S$ 
3:  $S \leftarrow \emptyset$ 
4: Select a vertex  $v \in V$  at random as the first facility and add it to  $S$ , i.e.,  $S \leftarrow S \cup \{v\}$ 
5: while  $|S| < p$  do
6:   Select the client vertex  $w$  ( $w \in V, w \notin S$ ) that has the largest distances to its facility  $w_f$ 
7:   if  $\text{rand}(0, 1) < \alpha$  then
8:     Identify  $N_{wk}$  for vertex  $w$  according to  $w_f$ 
9:     Select a vertex  $v$  from  $N_{wk}$  ( $v \notin S$ ) at random and add it to  $S$ , i.e.,  $S \leftarrow S \cup \{v\}$ 
10:  else
11:    Select a vertex  $v$  from  $N_w$  ( $v \notin S$ ) at random and add it to  $S$ , i.e.,  $S \leftarrow S \cup \{v\}$ 
12:  end if
13:  Identify the closet facility for all the vertices in  $V$ 
14: end while
15: return  $S$ 

```

mover troca (i, j) é desempenhado. A pesquisa tabu sempre seleciona um movimento não tabulado em cada iteração que minimiza o valor do objetivo ao máximo. O status tabu de um movimento é negligenciado apenas se o movimento levar a uma nova solução melhor do que a melhor solução encontrada até agora, ou se todos os movimentos estiverem em status tabu. Nossa pesquisa tabu para quando atinge um número máximo de iterações

eu_{max} (chamada de profundidade da pesquisa tabu).

2.4 Revinculação de caminho

Path-relinking (PR) foi originalmente proposto por Glover [38] como uma estratégia de intensificação explorando trajetórias conectando soluções de elite obtidas por busca tabu ou busca dispersa [39-41]. Partindo de uma ou mais soluções de elite, caminhos no espaço de soluções que levam a outras soluções de elite são gerados e explorados na busca por melhores soluções. Para gerar caminhos, movimentos são selecionados para introduzir atributos que estão contidos na solução de orientação de elite na solução atual. A reconexão de caminhos pode ser vista como uma estratégia que busca incorporar atributos de soluções de alta qualidade, favorecendo esses atributos nas jogadas selecionadas.

Algoritmo 3 dá a implementação do pathrelinking para o p -problema do centro. Deixar S_{c1} e S_{e1} sejam os conjuntos que contêm vértices diferentes entre S_c e S_e , respectivamente. Deixar Q ser um conjunto que contém triádes definidas como (S, v_{eu}, v_j) , Onde S é uma solução para o problema, e v_{eu} e v_j são dois vértices diferentes. Para cada vértice v_c dentro S_{c1} e cada vértice v_e dentro S_{e1} , remove v_c fora de S_r e adiciona v_e para dentro S_r . A solução resultante S_t e os vértices correspondentes v_c e v_e são compostos como um elemento (S_t, v_c, v_e) que é adicionado a Q

(linha 8). A melhor solução no elemento (S, v_{eu}, v_j) de Q (ou seja, a solução com o menor incremento do ob-valor da função jetiva) é registrado em S_r (linha 11). Então remove v_c e v_e fora de S_{c1} e S_{e1} , respectivamente. Este procedimento se repete $\beta \times d(S_c, S_e)$ tempos onde β é um parâmetro que representa a proporção da posição de S_r no caminho da solução inicial para a solução orientadora. Depois, a solução resultante S_r é otimizado pelo procedimento de pesquisa tabu.

β é um parâmetro importante que controla a distância entre a solução intermediária e a solução inicial. Representa o efeito de diversificação da reconexão de caminhos. É razoável considerar que um valor muito grande ou pequeno de β pode não fornecer uma dose adequada de diversificação porque produz uma solução que está muito próxima da solução guia ou da solução inicial. Conduzimos experimentos extensivos para analisar a influência do parâmetro β sobre o desempenho de GRASP / PR na Subseção 2.4.

3 Resultados Experimentais e Comparações

Nesta seção, relatamos resultados experimentais abrangentes da aplicação de GRASP / PR nas instâncias de benchmark padrão bem conhecidas e comparamos o desempenho de GRASP / PR com os algoritmos de última geração na literatura.

3.1 Instâncias do problema e protocolo experimental

Realizamos experimentos computacionais em dois conjuntos de problemas de teste. O primeiro conjunto é o p - instâncias de problemas não capacitados medianos retirados do OR-

Algorithm 3. Pseudo-Code of Path-Relinking for p -Center

- 1: **Input:** the current GRASP solution S^c , the guiding solution S^e , parameter β
 - 2: **Output:** the best solution S^r found by path-relinking
 - 3: $S^r \leftarrow S^c, Q \leftarrow \emptyset$
 - 4: $S^{c1} \leftarrow S^c \setminus (S^c \cap S^e), S^{e1} \leftarrow S^e \setminus (S^c \cap S^e)$
 - 5: **for** i from 1 to $\beta \times d(S^c, S^e)$ **do**
 - 6: **for** each vertex $v_c \in S^{c1}$ **do**
 - 7: **for** each vertex $v_e \in S^{e1}$ **do**
 - 8: $S^t \leftarrow S^r \setminus \{v_c\}, S^t \leftarrow S^r \cup \{v_e\}, Q \leftarrow Q \cup \{(S^t, v_c, v_e)\}$
 - 9: **end for**
 - 10: **end for**
 - 11: $(S^r, v_c, v_e) \leftarrow \operatorname{argmin} \{f(S) | (S, v_i, v_j) \in Q\}$
 - 12: $S^{c1} \leftarrow S^{c1} \setminus \{v_c\}, S^{e1} \leftarrow S^{e1} \setminus \{v_e\}, Q \leftarrow \emptyset$
 - 13: **end for**
 - 14: $S^r \leftarrow \text{TabuSearch}(S^r)$
-

Biblioteca^③. O conjunto de referência consiste em 40 instâncias com $|V|$ variando de 100 a 900 e p variando de 5 a 90. O segundo conjunto são as instâncias TSP do TSPLIB^④.

Este conjunto consiste em 84 instâncias com $|V|$ variando de 226 a 1817 e p variando de 5 a 150. Uma descrição detalhada das instâncias pode ser encontrada em [42]. Em todos os casos, a matriz de menor distância d não é fornecido diretamente, portanto, inicializamos a matriz d com o algoritmo Floyd [43] para as instâncias na biblioteca OR que fornece os comprimentos das arestas e distâncias euclidianas para as instâncias no TSPLIB que fornece as coordenadas dos vértices.

Nosso algoritmo foi programado em C++ e executado em um PC com CPU de 3,40 GHz e 4 GB de RAM. Conduzimos experimentos preliminares para encontrar os melhores valores de todos os parâmetros usados em nosso algoritmo em todas as instâncias testadas. Além disso, apresentamos uma análise detalhada e ajuste de dois parâmetros importantes (α , β) na Subseção 4.2. No seguinte experimento, nós definimos eu_{max} , tt , L_{max} , α , β , e o tamanho máximo do conjunto de elite P a 10.000, $p \times (|V| - p) / 100 + rand(10 \times p)$, 10.000, 0,7, 0,5 e 10, respectivamente. Todos os resultados computacionais são obtidos sem ajuste especial dos parâmetros. Executamos GRASP / PR em cada instância do problema por 20 execuções independentes e comparamos com vários algoritmos de última geração na literatura.

3.2 Resultados Computacionais

Comparamos o desempenho do GRASP / PR com três algoritmos exatos (ELP, IP* e Daskin*) e dois algoritmos metaheurísticos (VNS e PBS). ELP foi executado em um PC com uma CPU Pentium II de 400 MHz e 384 MB de RAM [8]. IP* e Daskin* foram executados em uma Sun Enterprise Workstation 450 [9]. O VNS foi executado em uma estação Sun Sparc 10. O único parâmetro do VNS, ou seja, o número de estruturas de vizinhança, é definido como p [13]. O PBS foi executado em um PC com CPU AMD Opteron 252 2,6 GHz [13]. No entanto, uma comparação completamente justa é impossível, uma vez que temos uma abordagem não determinística (ou seja, nossa abordagem) por um lado e as abordagens determinísticas por outro. Além disso, diferentes ambientes de computação constituem outra fonte importante de dificuldade para uma comparação justa. Portanto, os tempos computacionais relatados são dados apenas para fins indicativos.

De acordo com o tamanho de V , classificamos as instâncias de teste em duas categorias: pequenas instâncias ($|V| <$

1 000) e grandes instâncias ($|V| > 1 000$). As pequenas instâncias incluem os 40 p - instâncias medianas e 11 instâncias do TSPLIB: "pr226, pr264, pr299, pr439, pcb442, kroA200, kroB200, lin318, gr202, d493, d657". As grandes instâncias são "u1060, rl1323, u1817".

3.2.1 Resultados Computacionais das Pequenas Instâncias

A Tabela 1 e a Tabela 2 relatam os resultados computacionais do algoritmo proposto em pequenas instâncias. Especificamente, a Tabela 1 apresenta os resultados computacionais de 40 p - instâncias medianas e faz comparação com ELP, IP*, Daskin*, VNS e PBS, onde coluna t (s)

mostra o tempo de CPU para ELP, IP* e Daskin* e o tempo médio de CPU para VNS, PBS e GRASP / PR, e coluna t_{std} em PBS e GRASP / PR mostra o desvio padrão do tempo de CPU para cada instância.

A Tabela 2 apresenta as pequenas instâncias do TSPLIB e faz uma comparação com IP*, Daskin* e PBS. Col-umn f_{melhor} apresenta o melhor valor de solução obtido por PBS e GRASP / PR. Como PBS e GRASP / PR usar dados de ponto flutuante e os algoritmos exatos de referência usam dados inteiros, um resultado obtido por PBS ou GRASP / PR é considerado idêntico ao resultado obtido pelos algoritmos exatos se o resultado arredondado de PBS ou GRASP / PR for igual ao resultado obtido pelo algoritmo exato.

Da Tabela 1 e Tabela 2, observa-se que, embora GRASP / PR e PBS possam obter as soluções ideais para todas as instâncias na biblioteca OR e TSPLIB com uma taxa de sucesso de 100%, GRASP / PR supera PBS e os outros algoritmos para tem o menor tempo computacional médio. Além do mais, GRASP / PR tem um valor menor de t_{std} em comparação com PBS. Além disso, para instâncias pr299 com 40 instalações e gr202 com 6 facilidades, GRASP / PR pode obter um resultado relativamente melhor do que PBS (indicado em negrito).

3.2.2 Resultados Computacionais de Grandes Instâncias

Nesta subseção, conduzimos experimentos de GRASP / PR nas grandes instâncias com até 1.817 vértices e 150 instalações. A Tabela 3 apresenta os resultados computacionais de GRASP / PR e faz a comparação com PBS, ELP e VNS nas grandes instâncias. Col-umn f_{optar} mostra os valores de solução ideais obtidos por ELP, onde um valor de solução ideal que não é identificado marcada pela ELP é marcada com "?". Coluna f_{melhor} mostra o melhor valor de solução obtido pelo referenciado al-gorismo. Colunas t (s), t_{std} , e f_{dev} mostra a média

^③<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/pmedinfo.html>, Jan. 2017.

^④<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/>, Jan. 2017.

Tabela 1. Resultados computacionais e comparação com os algoritmos de referência em pequenas instâncias da biblioteca OR

Instance	n	p	Density	f_{opt}	$t(s)$				PBS		GRASP/PR	
					ELP	IP*	Daskin*	VNS	$t(s)$	t_{std}	$t(s)$	t_{std}
pmed01	100	5	0.040 4	127	0.70	4.05	2.09	0.04	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$
pmed02	100	10	0.040 4	98	0.20	1.52	1.45	4.45	0.01	0.01	$< \varepsilon$	$< \varepsilon$
pmed03	100	10	0.040 4	93	0.10	1.81	1.35	0.17	0.06	0.05	0.01	0.02
pmed04	100	20	0.040 4	74	0.10	1.01	0.92	0.30	$< \varepsilon$	0.01	$< \varepsilon$	0.01
pmed05	100	33	0.040 4	48	0.10	1.49	0.73	0.11	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$
pmed06	200	5	0.040 2	84	0.30	13.53	9.01	1.51	0.02	0.02	$< \varepsilon$	$< \varepsilon$
pmed07	200	10	0.040 2	64	0.50	5.09	4.31	1.30	0.01	0.02	$< \varepsilon$	$< \varepsilon$
pmed08	200	20	0.040 2	55	0.40	5.31	3.34	2.22	0.01	0.01	$< \varepsilon$	0.01
pmed09	200	40	0.040 2	37	0.10	3.46	2.66	10.89	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$
pmed10	200	67	0.040 2	20	0.30	2.76	2.57	9.04	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$
pmed11	300	5	0.040 1	59	1.00	11.67	16.25	1.52	0.04	0.06	0.01	0.03
pmed12	300	10	0.040 1	51	1.30	12.03	12.25	5.39	0.01	0.02	$< \varepsilon$	$< \varepsilon$
pmed13	300	30	0.040 1	36	0.80	14.43	8.23	10.21	0.05	0.05	$< \varepsilon$	0.02
pmed14	300	60	0.040 1	26	0.90	6.61	6.81	70.67	0.01	0.01	$< \varepsilon$	$< \varepsilon$
pmed15	300	100	0.040 1	18	1.00	4.43	4.40	55.86	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$
pmed16	400	5	0.040 1	47	1.60	30.01	28.10	0.08	0.01	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$
pmed17	400	10	0.040 1	39	2.10	30.88	27.06	26.00	0.02	0.02	$< \varepsilon$	0.02
pmed18	400	28	0.040 1	28	1.40	12.49	13.17	119.61	0.13	0.12	0.01	0.07
pmed19	400	80	0.040 1	18	0.40	9.89	10.16	300.11	1.08	1.01	0.57	0.93
pmed20	400	133	0.040 1	13	1.80	13.53	9.30	218.61	0.10	0.08	0.03	0.07
pmed21	500	5	0.040 1	40	5.20	56.40	56.01	1.24	0.01	0.01	0.01	0.01
pmed22	500	10	0.040 1	38	4.30	495.20	60.78	82.63	1.12	1.18	0.24	1.02
pmed23	500	50	0.040 1	22	1.20	28.52	16.45	112.33	2.11	1.84	0.14	1.43
pmed24	500	100	0.040 1	15	4.50	14.64	12.59	264.46	0.06	0.04	0.04	0.03
pmed25	500	167	0.040 1	11	2.70	13.06	10.28	175.67	0.05	0.04	0.02	0.02
pmed26	600	5	0.040 1	38	6.10	401.60	104.20	0.58	0.03	0.03	$< \varepsilon$	$< \varepsilon$
pmed27	600	10	0.040 1	32	8.20	78.24	65.23	5.07	0.04	0.04	$< \varepsilon$	0.04
pmed28	600	60	0.040 1	18	2.10	39.51	19.31	25.09	0.13	0.11	0.04	0.05
pmed29	600	120	0.040 1	13	5.10	32.00	23.61	762.44	0.05	0.03	0.03	0.01
pmed30	600	200	0.040 1	9	5.40	34.72	17.22	196.95	0.80	0.60	0.35	0.23
pmed31	700	5	0.040 1	30	8.10	303.00	122.60	0.58	0.03	0.01	$< \varepsilon$	$< \varepsilon$
pmed32	700	10	0.040 1	29	45.20	1 447.00	116.80	165.26	0.31	0.44	0.05	0.31
pmed33	700	70	0.040 1	15	3.10	94.07	33.11	806.77	81.75	107.25	1.56	2.56
pmed34	700	140	0.040 1	11	6.50	50.23	29.66	160.15	0.04	0.01	0.04	0.01
pmed35	800	5	0.040 1	30	13.70	183.90	123.30	6.67	0.10	0.12	0.04	0.10
pmed36	800	10	0.040 1	27	34.50	3 602.00	110.50	105.99	0.96	0.97	0.55	0.68
pmed37	800	80	0.040 1	15	2.00	105.80	49.02	1 197.86	0.27	0.21	0.08	0.13
pmed38	900	5	0.031 6	29	18.50	251.00	273.10	1.92	0.03	0.01	0.03	$< \varepsilon$
pmed39	900	10	0.040 0	23	27.30	5 817.00	208.70	5.98	26.30	27.12	0.77	1.44
pmed40	900	90	0.040 0	13	7.80	240.80	462.90	493.79	0.46	0.29	0.25	0.27
Avg.					5.67	336.87	51.99		2.91	3.55	0.12	0.24

Mesa 2. Resultados computacionais e comparação com os algoritmos de referência em pequenas instâncias do TSPLIB

Instance	n	p	f_{opt}	f_{best}		$t(s)$		PBS		GRASP/PR	
				PBS	GRASP/PR	IP*	Daskin*	$t(s)$	t_{std}	$t(s)$	t_{std}
pr226	226	40	650	650.00	650.00	7.39	5.68	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$
pr226	226	20	1 366	1 365.65	1 365.65	9.55	7.28	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$
pr226	226	10	2 326	2 326.48	2 326.48	8.85	9.29	0.03	0.03	0.010	0.02
pr226	226	5	3 721	3 720.55	3 720.55	19.92	8.08	0.01	0.02	$< \varepsilon$	0.02
pr264	264	40	316	316.23	316.23	-	-	$< \varepsilon$	0.01	$< \varepsilon$	0.01
pr264	264	20	515	514.78	514.78	11.19	9.50	0.02	0.02	$< \varepsilon$	0.01
pr264	264	10	850	850.00	850.00	12.00	11.73	0.01	0.01	$< \varepsilon$	$< \varepsilon$
pr264	264	5	1 610	1 610.12	1 610.12	14.62	16.27	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$
pr299	299	40	355	355.52	355.32	12.95	11.70	0.18	0.16	0.050	0.08
pr299	299	20	559	559.02	559.02	14.29	13.10	0.22	0.23	0.140	0.15
pr299	299	10	889	888.84	888.84	23.95	18.62	0.43	0.36	0.300	0.33
pr299	299	5	1 336	1 336.27	1 336.27	18.24	18.74	0.09	0.11	0.050	0.02
pr439	439	40	672	671.75	671.75	32.52	33.03	0.66	0.76	0.270	0.34
pr439	439	20	1 186	1 185.59	1 185.59	35.11	29.21	0.12	0.14	0.020	0.05
pr439	439	10	1 972	1 971.83	1 971.83	39.96	36.16	0.09	0.12	0.010	0.09
pr439	439	5	3 197	3 196.58	3 196.58	53.08	51.74	1.84	2.02	0.040	0.06
pcb442	442	40	316	316.23	316.23	2 714.00	1 302.00	0.04	0.04	0.020	0.04
pcb442	442	20	447	447.21	447.21	37.62	151.20	0.89	0.95	0.230	0.11
pcb442	442	10	671	670.82	670.82	34.30	128.80	0.27	0.33	0.060	0.26
pcb442	442	5	1 025	1 024.74	1 024.74	40.31	41.37	0.47	0.52	0.080	0.18
kroA200	200	40	258	258.26	258.26	5.19	5.02	0.14	0.12	0.610	0.09
kroA200	200	20	389	389.31	389.31	5.51	5.38	0.12	0.11	0.050	0.09
kroA200	200	10	599	598.82	598.82	6.57	6.83	0.58	0.61	0.110	0.23
kroA200	200	5	911	911.41	911.41	8.22	8.28	0.10	0.12	0.020	0.11
kroB200	200	40	253	253.24	253.24	5.08	4.29	0.04	0.04	0.010	0.01
kroB200	200	20	382	382.28	382.28	8.27	4.75	0.04	0.03	$< \varepsilon$	$< \varepsilon$
kroB200	200	10	582	582.10	582.10	6.69	5.84	0.05	0.05	0.001	0.03
kroB200	200	5	898	897.67	897.67	7.63	7.53	0.01	0.01	$< \varepsilon$	$< \varepsilon$
lin318	318	40	316	315.92	315.92	219.40	11.67	0.01	0.01	$< \varepsilon$	$< \varepsilon$
lin318	318	20	496	496.45	496.45	13.54	14.49	15.92	20.18	0.860	1.34
lin318	318	10	743	743.21	743.21	17.46	18.52	0.47	0.50	0.060	0.38
lin318	318	5	1 101	1 101.34	1 101.34	18.68	22.06	0.26	0.35	0.050	0.12
gr202	202	40	3	2.97	2.97	5.15	3.81	0.05	0.05	$< \varepsilon$	0.02
gr202	202	20	6	5.97	5.57	5.65	4.63	0.01	0.01	$< \varepsilon$	$< \varepsilon$
gr202	202	10	9	9.33	9.33	6.79	4.89	0.05	0.07	$< \varepsilon$	$< \varepsilon$
gr202	202	5	19	19.38	19.38	8.59	5.83	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$	$< \varepsilon$
d493	493	40	206	206.02	206.02	45.38	79.97	36.25	32.78	6.950	5.96
d493	493	20	313	312.74	312.74	1 406.00	75.54	19.36	21.75	2.580	8.23
d493	493	10	458	458.30	458.30	60.13	82.25	5.21	4.79	1.270	1.21
d493	493	5	753	752.91	752.91	72.24	75.41	19.96	15.86	3.490	2.74
d657	657	40	250	249.52	249.52	3 126.00	351.70	196.90	354.34	22.340	7.36
d657	657	20	375	374.70	374.70	301.15	255.80	5.30	5.07	0.460	4.55
d657	657	10	575	574.74	574.74	751.01	156.60	28.58	25.70	4.030	3.31
d657	657	5	881	880.91	880.91	100.56	154.70	209.97	28.66	0.610	1.27
Avg.						212.52	74.30	12.38	11.75	1.020	0.88

Tabela 3. Resultados computacionais e comparação com os algoritmos de referência em grandes instâncias do TSPLIB

Instance	n	p	ELP		f_{best}		PBS			GRASP/PR			VNS		
			f_{opt}	$t(s)$	PBS	GRASP/PR	$t(s)$	t_{std}	f_{dev}	$t(s)$	t_{std}	f_{dev}	f_{best}	$t(s)$	f_{dev}
u1060	1060	10	2 273	53	2 273.08	2 273.08	138.11	56.30	0.00	1.31	24.11	0.00	2 280.09	94.93	0.31
u1060	1060	20	1 581	2 778	1 580.80	1 580.80	659.44	1 449.55	0.00	14.88	85.67	0.00	1 611.95	20.49	1.07
u1060	1060	30	1 208	298	1 207.77	1 207.77	36.84	25.89	0.00	3.19	30.43	0.00	1 220.41	373.46	0.24
u1060	1060	40	1021	366	1 020.56	1 020.56	47.73	43.65	0.00	3.26	41.32	0.00	1 050.45	279.75	2.93
u1060	1060	50	905	383	904.92	904.92	233.13	129.32	0.00	218.85	104.87	0.00	922.14	477.18	0.00
u1060	1060	60	781	233	781.17	781.17	103.12	74.46	0.00	7.75	89.55	0.00	806.52	446.89	3.25
u1060	1060	70	711	135	710.76	710.75	109.56	20.23	0.00	116.91	12.40	0.00	721.37	422.73	1.49
u1060	1060	80	652	60	652.16	652.16	142.11	54.56	0.00	316.57	38.53	0.00	670.53	398.84	2.81
u1060	1060	90	608	38	607.88	607.87	63.15	27.14	0.00	7.09	20.78	0.00	640.23	111.08	5.32
u1060	1060	100	570	29	570.01	570.01	17.54	16.43	0.00	19.04	8.29	0.00	582.92	430.33	2.26
u1060	1060	110	539	30	538.84	538.84	160.73	74.35	0.00	66.46	57.33	0.00	565.72	186.60	4.99
u1060	1060	120	510	44	510.28	510.27	107.65	28.90	0.00	397.85	18.70	0.00	551.90	218.84	8.16
u1060	1060	130	500	44	499.65	499.65	118.71	77.53	0.00	58.18	83.08	0.00	500.14	473.65	0.10
u1060	1060	140	452	46	452.46	452.46	318.48	150.04	0.00	127.39	55.86	0.00	500.12	214.06	10.37
u1060	1060	150	447	50	447.01	447.01	10.59	12.87	0.00	4.37	11.50	0.00	453.16	428.16	1.38
rl1323	1323	10	3 077	1 380	3 077.30	3 077.30	4 760.10	1 905.62	0.00	38.02	342.59	0.00	-	-	-
rl1323	1323	20	2 016	480	2 016.40	2 016.40	605.90	226.38	0.00	104.89	129.04	0.00	-	-	-
rl1323	1323	30	1 632	900	1 631.50	1 631.50	1 200.20	1 138.23	0.00	169.47	473.51	0.00	-	-	-
rl1323	1323	40	1 352	3 000	1 352.36	1 352.36	292.00	154.02	0.00	21.90	184.90	0.00	-	-	-
rl1323	1323	50	1 187	8 580	1 187.27	1 187.27	619.40	211.66	0.00	119.63	110.75	0.00	-	-	-
rl1323	1323	60	1 063	9 120	1 063.01	1 063.01	8 184.90	8 206.12	0.03	4 190.92	394.07	0.00	-	-	-
rl1323	1323	70	972	1 740	971.93	971.93	7 427.00	4 157.30	0.03	6 287.04	129.23	0.00	-	-	-
rl1323	1323	80	895	420	895.06	895.06	8 783.00	6 698.80	0.28	5 265.81	384.50	0.09	-	-	-
rl1323	1323	90	832	120	832.00	832.00	929.90	584.37	0.00	776.23	453.29	0.00	-	-	-
rl1323	1323	100	787	120	789.70	789.70	1 840.50	1 070.96	0.34	2 010.67	225.68	0.00	-	-	-
u1817	1817	10	458	2 700	457.91	457.91	5 316.60	728.10	0.00	604.53	87.25	0.00	-	-	-
u1817	1817	20	3 10?	4 920	309.01	309.01	10 243.00	3 842.53	0.00	4 068.06	398.87	0.00	-	-	-
u1817	1817	30	2 50?	16 500	240.99	240.99	1 605.50	371.02	0.00	1 239.97	20.43	0.00	-	-	-
u1817	1817	40	2 10?	6 420	209.46	209.45	193.70	206.73	0.00	308.29	67.34	0.00	-	-	-
u1817	1817	50	1 87?	9 840	184.91	184.91	1 128.90	714.66	0.00	471.94	234.90	0.00	-	-	-
u1817	1817	60	163	1 260	162.65	162.64	837.30	287.25	0.00	469.43	55.98	0.00	-	-	-
u1817	1817	70	148	420	148.11	148.11	191.80	193.45	0.00	19.66	87.36	0.00	-	-	-
u1817	1817	80	137	1 140	136.80	136.80	127.50	105.50	0.00	12.42	110.40	0.00	-	-	-
u1817	1817	90	1 30?	7 202	129.54	129.51	2 963.50	2 646.63	0.00	3 859.05	287.61	0.00	-	-	-
u1817	1817	100	127	300	127.01	126.99	146.40	142.66	0.00	2.35	39.51	0.00	-	-	-
u1817	1817	110	109	420	109.25	109.25	13 772.40	16 610.21	0.00	6 954.89	434.03	0.00	-	-	-
u1817	1817	120	108	120	107.78	107.76	80.10	62.16	0.00	5.25	19.34	0.00	-	-	-
u1817	1817	130	1 08?	3 720	107.75	107.75	11.20	6.46	0.00	7.04	13.45	0.00	-	-	-
u1817	1817	140	1 05?	4 020	101.61	101.60	4 949.30	2 349.44	0.00	30.95	137.31	0.00	-	-	-
u1817	1817	150	94?	5 640	101.60	92.44	314.00	54.35	0.00	1 236.55	23.97	0.16	-	-	-
Avg.				2 376.73			1 969.78	1 372.90	0.02	990.95	138.19	0.01	-	-	-

tempo computacional em segundos, o desvio padrão do tempo computacional e o desvio médio do valor da solução do melhor valor da solução encontrado, respectivamente. O resultado que não é relatado pelo VNS em [13] é marcado com "-".

Da Tabela 3, observa-se que GRASP / PR pode melhorar os resultados anteriores mais conhecidos para 10 casos em comparação com PBS (u1060 com $p = 70, 90$ e $u1817$ com $p = 40, 60, 80, 90, 100, 120$ e 150) e corresponda aos resultados anteriores mais conhecidos para os outros. Além disso, GRASP / PR supera o PBS por possuir valores menores de tempo médio computacional, desvio padrão do tempo computacional e desvio do valor da solução mais conhecida. Comparado com VNS em instâncias u1060, GRASP / PR tem a superioridade de qualidade de solução e computação

eficiência internacional, pois tem valores menores de f_{melhor} , $t(s)$, e f_{dev} . Observe que GRASP / PR pode obter os mesmos valores de solução com 100% de taxa de sucesso para 20 independentes dent é executado para todas as instâncias, exceto r1323 com $p = 80$ e $u1817$ com $p = 150$. Além disso, por exemplo, $u1817$ com $p = 150$, o PBS não consegue alcançar o melhor resultado obtido por ELP, enquanto GRASP / PR pode obter um resultado melhor do que ELP. Esses resultados demonstram o bom desempenho do GRASP / PR em termos de qualidade da solução e eficiência computacional.

4 Análise e Discussões

4.1 Importância da Estratégia Tabu

Para avaliar o desempenho da busca tabu no algoritmo GRASP / PR, modificamos GRASP / PR substituindo o procedimento de busca tabu pelo procedimento de busca local introduzido em [23]. O novo algoritmo é denotado por GRASP / PR-I. Aplicamos GRASP / PR e GRASP / PR-I em cada uma das instâncias u1817 com 10 ~ 150 instalações para 20 execuções independentes e relatar os resultados comparativos na Tabela 4, onde a coluna acertar mostra a taxa de sucesso para alcançar o melhor

resultado conhecido f_{melhor} mais de 20 corridas.

Da Tabela 4, observa-se que GRASP / PR supera GRASP / PR-I por poder obter melhores resultados para oito casos (ou seja, $p = 40, 60, 90, 100, 120, 130, 140$ e 150), e o tempo médio computacional e o desvio padrão do tempo computacional de GRASP / PR são menores do que os de GRASP / PR-I. Além disso, GRASP / PR é mais estável que GRASP / PR-I por apresentar maior taxa de acerto e menor desvio médio da melhor solução encontrada. Estes indicam a eficácia da pesquisa tabu como um procedimento de pesquisa local no GRASP / PR para o p- problema do centro.

Tabela 4. Comparação entre GRASP / PR e GRASP / PR-I na instância u1817 com 10 ~ 150 instalações

Instance	n	p	GRASP/PR-I					GRASP/PR				
			f_{best}	f_{dev}	$t(s)$	t_{std}	hit	f_{best}	f_{dev}	$t(s)$	t_{std}	hit
u1817	1817	10	457.91	0.00	2084.47	384.90	20/20	457.91	0.00	604.53	87.25	20/20
u1817	1817	20	309.01	0.00	5672.01	1023.84	20/20	309.01	0.00	4068.06	398.87	20/20
u1817	1817	30	240.99	0.00	3048.76	304.85	20/20	240.99	0.00	1239.97	20.43	20/20
u1817	1817	40	209.46	0.59	1093.80	128.53	16/20	209.45	0.00	308.29	67.34	20/20
u1817	1817	50	184.91	0.12	1395.41	582.49	18/20	184.91	0.00	471.94	234.90	20/20
u1817	1817	60	162.65	0.48	763.55	243.56	15/20	162.64	0.00	469.43	55.98	20/20
u1817	1817	70	148.11	0.00	137.49	114.70	20/20	148.11	0.00	19.66	87.36	20/20
u1817	1817	80	136.80	0.00	88.73	95.11	20/20	136.80	0.00	12.42	110.40	20/20
u1817	1817	90	129.54	0.22	2014.87	1645.83	13/20	129.51	0.00	3859.05	287.61	20/20
u1817	1817	100	127.01	0.00	76.48	124.65	20/20	126.99	0.00	2.35	39.51	20/20
u1817	1817	110	109.25	0.00	7723.49	1395.20	20/20	109.25	0.00	6954.89	434.03	20/20
u1817	1817	120	107.78	0.00	57.66	164.09	20/20	107.76	0.00	5.25	19.34	20/20
u1817	1817	130	107.76	0.28	33.98	14.58	15/20	107.75	0.00	7.04	13.45	20/20
u1817	1817	140	102.39	0.00	2487.76	879.34	20/20	101.60	0.00	30.95	137.31	20/20
u1817	1817	150	101.60	0.32	302.54	47.56	12/20	92.44	0.16	1236.55	23.97	17/20
Avg.				0.13	1798.73	476.62	17.27/20		0.01	1286.03	134.52	19.820

4.2 Influência dos Parâmetros α e β

Conforme indicado na Seção 2, parâmetro α controla a quantidade de ganância e aleatoriedade no procedimento de solução inicial e parâmetro β representa o efeito de diversificação da reconexão de caminhos. Nesta subseção, analisamos a influência dos parâmetros α e β sobre o desempenho do GRASP / PR. Para parâmetro α , pegamos 11 valores diferentes $\alpha \in [0, 1]$ (tamanho do passo de 0,1) e mantenha os outros parâmetros fixos. Para parâmetro β , tomamos 21 valores diferentes de $\beta \in [0, 1]$ (tamanho do passo de 0,05) e mantenha os outros parâmetros fixos. Realizamos 20 execuções independentes para cada parâmetro e cada uma das 15 grandes instâncias u1060 com 10 ~ 150 instalações, e pare nosso algoritmo quando atingir 5.000 iterações. Fig.1 (a) mostra o valor objetivo normalizado médio \circ do soluções iniciais (obtidas pelo procedimento de solução inicial, denotadas por "norm.f de solução inicial") e as soluções finais (obtidas por GRASP / PR, denotadas por "norm.f solução final") correspondendo a diferentes valores de parâmetro α . A Fig.1 (b) mostra o valor objetivo normalizado médio e o tempo computacional, e sua curva de ajuste quadrática correspondendo a diferentes valores de parâmetro β .

Da Fig.1 (a), observa-se que o valor objetivo médio da solução inicial aumenta gradativamente com o incremento de α de 0 a 1, enquanto o valor objetivo médio da solução final não muda muito quando $\alpha \in [0, 1]$. Isso implica que α pode influenciar a qualidade da solução inicial, mas tem pouco impacto na qualidade da solução final de GRASP / PR. Considerando que o valor objetivo médio da solução final é o menor quando $\alpha = 0,7$, α é sugerido ser 0,7.

Da Fig.1 (b), pode-se observar que GRASP / PR pode encontrar uma solução de qualidade relativamente alta quando $\beta \in [0,35, 0,7]$, enquanto o tempo computacional médio correspondente aumenta quando $\beta \in [0, 0,65]$ e diminui quando $\beta \in [0,65, 1]$. Isso indica que, para o parâmetro β , apenas os valores no meio da região $[0, 1]$ podem introduzir uma diversificação considerável na pesquisa, mas requer mais tempo computacional. A razão pode ser que, se β é muito grande ou pequeno, as soluções intermediárias estão muito próximas da solução inicial ou orientadora e, portanto, é fácil para a pesquisa tabu ficar presa nos ótimos locais anteriores. Para testemunhar

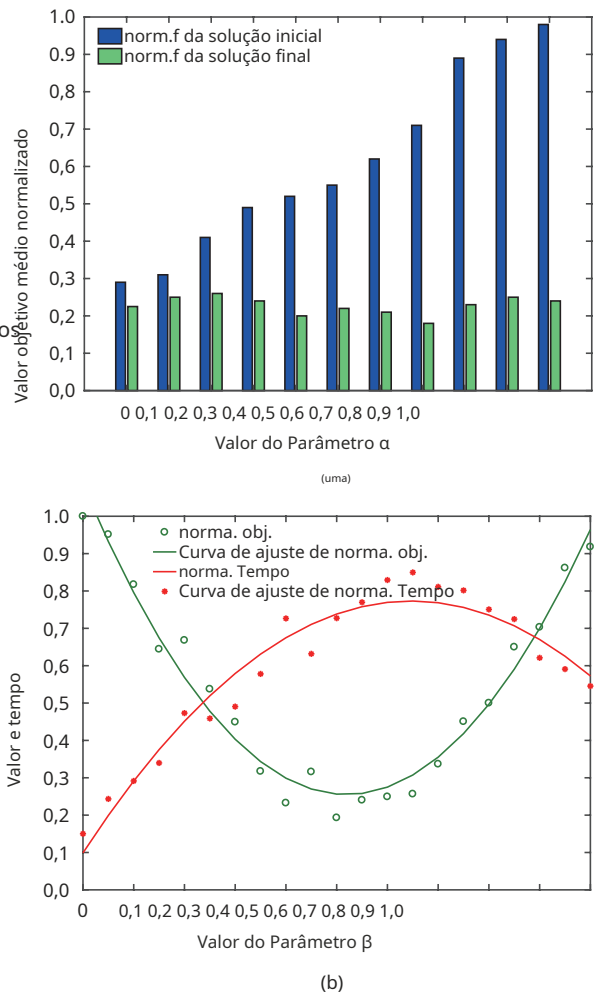


Figura 1. Valor médio normalizado da função objetivo e tempo computacional correspondente a diferentes valores de parâmetros α e β . norma. significa valor normalizado médio e obj. é a abreviação de objetivo.

este fenômeno, aplicamos GRASP / PR nas 15 grandes instâncias u1060 com 10 ~ 150 instalações com 11 valores diferentes de $\beta \in [0, 1]$ (tamanho do passo de 0,1) e registre o número de vezes que a solução retornada de pathrelinking encontra (é o mesmo que) os ótimos locais visitados anteriormente (aqui os ótimos locais são soluções obtidas por pesquisa tabu). Os resultados são apresentados na Tabela 5, onde a coluna "Valor de β " representa o valor de β e a coluna "Número de Encontro" representa a média de vezes que a solução retornada de reconexão de caminho encontra o ótimo local visitado anteriormente. Da Tabela 5, observa-se que o número de encontros diminui quando $\beta \in [0, 0,5]$, embora em

③ Average normalized objective value: first, finding the maximum and minimum objective values (denoted by f_{\max} and f_{\min} respectively) over the 20 runs for each instance; second, normalizing the objective value of each run for each instance using $norm.f = (f - f_{\min}) / (f_{\max} - f_{\min} + 1)$; third, averaging the $norm.f$ for all the 15 instances for each value of parameter α or β .

vinca quando $\beta \in [0,5, 1]$. Isso confirma que o procedimento de pesquisa tabu na reconexão de caminhos é fácil de ser capturado em ótimos anteriores com um valor muito grande ou pequeno de β . Considerando a qualidade da solução e o tempo computacional, β sugere-se que seja 0,5.

Tabela 5. Média de tempos em que a solução devolvida de reconexão de caminho encontra o local visitado anteriormente
Optima em GRASP / PR em u1060 com 10 ~ 150 Instalações correspondentes a diferentes valores de β

Value of β	Number of Encounters ($\times 10^5$)
0.0	12.36
0.1	10.89
0.2	9.58
0.3	7.33
0.4	6.01
0.5	4.75
0.6	5.24
0.7	7.18
0.8	8.63
0.9	9.93
1.0	11.50

4.3 Fitness-Distance Correlation and Distribution of Local Optima

The fitness-distance correlation (FDC) coefficient $\rho^{[44]}$ is a well-known tool for landscape analysis and can provide useful indications about the problem hardness. The FDC captures the correlation between the fitness (solution quality) of a solution and its distance to the nearest global optimum (or best-known solution if no global optimum is available). For a minimization problem, $\rho = 1$ ideally indicates a perfect correlation between fitness and distance to the optimum, implying that improving the fitness reduces the distance to the global optimum. For landscape $-1 < \rho < 1$, there is virtually no correlation between fitness and distance, while for $\rho = -1$, there is no correlation at all, which means that using the fitness to guide the search towards global optimum may be misleading.

Table 6 reports the results of FDC analysis on instances rl1323 and u1817 with 10~150 facilities. For each instance, we run GRASP/PR on it for 30 minutes and collect the number of distinct local optima (column num_d_{lo}), the average distance between local optima (column avg_d_{lo}), the average distance between

Table 6. FDC Results on Instances rl1323 and u1817 with 10~150 Facilities

Instance	n	p	num_d_{lo}	avg_d_{lo}	avg_d_{lb}	avg_d_{be}	ρ
rl1323	1323	10	843	7.586 50	6.73	6.73	0.56
rl1323	1323	20	1 590	14.154 80	10.81	10.81	0.47
rl1323	1323	30	2 512	27.416 00	23.04	23.04	0.53
rl1323	1323	40	3 301	28.001 50	27.55	27.55	0.64
rl1323	1323	50	4 219	36.096 90	35.62	35.62	0.62
rl1323	1323	60	5 296	44.502 60	44.41	44.41	0.77
rl1323	1323	70	6 198	58.400 30	51.52	51.52	0.61
rl1323	1323	80	6 979	59.304 30	59.88	59.88	0.64
rl1323	1323	90	7 679	61.269 20	62.86	62.86	0.74
rl1323	1323	100	9 626	64.240 20	67.65	67.65	0.70
u1817	1817	10	428	7.582 29	7.39	7.39	0.55
u1817	1817	20	852	17.267 60	15.33	15.33	0.35
u1817	1817	30	1 260	24.961 90	23.85	19.49	0.48
u1817	1817	40	1 814	32.651 60	35.46	32.60	0.53
u1817	1817	50	2 325	44.891 60	42.57	39.74	0.56
u1817	1817	60	3 019	58.885 40	54.46	51.84	0.73
u1817	1817	70	3 494	68.638 80	64.12	66.39	0.59
u1817	1817	80	3 919	69.865 10	72.21	70.92	0.58
u1817	1817	90	4 068	79.169 90	80.70	81.15	0.64
u1817	1817	100	4 271	89.875 40	85.92	92.23	0.60
u1817	1817	110	4 779	95.303 80	93.80	96.38	0.80
u1817	1817	120	6 160	105.595 80	102.93	110.67	0.75
u1817	1817	130	6 853	121.976 10	113.70	119.95	0.24
u1817	1817	140	6 419	132.718 80	129.07	130.23	0.61
u1817	1817	150	6 614	138.355 90	140.49	138.86	0.15

ótimos locais e a solução mais próxima melhor encontrada (coluna média u1817 with 40, 90, 150 facilities) in Fig.2, where the same data used for estimating ρ is displayed graphically. Such plots have been used to estimate the distribution of local optima for a number of problems including TSP^[45], QAP^[46], graph partitioning^[47-49], and

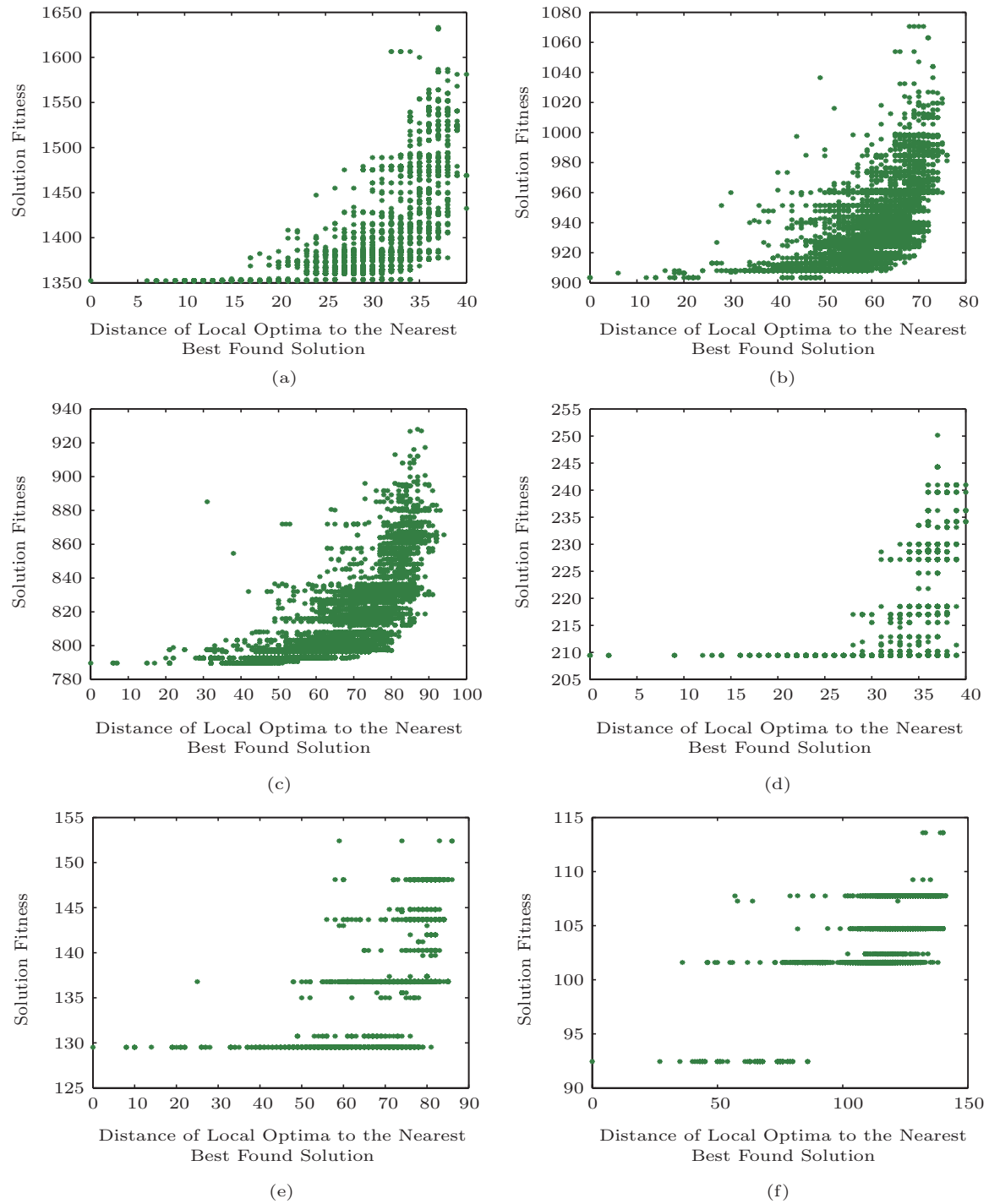


Fig.2. FDC plots of distance of local optima to the nearest best known solution for instances. (a) rl1323, $p = 40$. (b) rl1323, $p = 80$. (c) rl1323, $p = 100$. (d) u1817, $p = 40$. (e) u1817, $p = 90$. (f) u1817, $p = 150$.

problema de programação de fluxo de loja [50].

Da Tabela 6, observa-se que a maioria dos valores de d_{lo} , avg_d_{lo} , avg_d_{lb} , and avg_d_{be} are very large, close to the number of the facilities. This implies that local optima are scattered all over the search space. Besides, for the same number of facilities, the values of ρ of instances rl1323 are larger than these of instances u1817, which means that instances rl1323 with given facilities are comparably easy to solve. The FDC plots in Fig.2 also confirm this observation, where it can be seen that: on one hand, the distributions of local optima in rl1323 are denser than those in u1817; on the other hand, the distance of local optima to the nearest best found solution decreases when the solution fitness decreases in rl1323, while the distance of local optima to the nearest best found solution ranges in a wide region in u1817.

5 Conclusions

In this paper, we presented a GRASP/PR algorithm for solving the p -center problems which combines GRASP and path-relinking. The elite set of GRASP/PR consists of the local optimal solutions obtained by GRASP. Each iteration of GRASP consists of the construction of a randomized greedy solution, followed by a tabu search procedure. The resulting solution is combined with one of the elite solutions by path-relinking, which consists in exploring trajectories that connect high-quality solutions.

Tested on two sets of 124 well-known benchmarks, our GRASP/PR algorithm is competitive with the state-of-the-art algorithms in the literature in terms of both solution quality and computational efficiency. Specifically, it improves the previous best known results for 10 out of 40 large instances while matching the best known results for the others. In addition, the computational results demonstrate the robustness and computational efficiency of our GRASP/PR algorithm.

In addition, we investigated some essential ingredients of the proposed algorithm. First, we carried experiments to demonstrate the effectiveness of the tabu search as the local search procedure in GRASP/PR. Second, we analyzed the influence of parameters α and β on the performance of GRASP/PR and suggested a proper value for it.

The success of the GRASP/PR algorithm on the p -center problem reminds us that it is essential to introduce a mechanism to combine the independent elite solutions obtained by GRASP. Given the merits of

GRASP with path-relinking, we hope to design even more robust and effective heuristic algorithms for solving the p -center problem and other similar optimization problems such as capacitated p -center, p -median, maximum covering problems, and so on.

References

- [1] Kariv O, Hakimi S L. An algorithmic approach to network location problems. I: The p -centers. *SIAM Journal on Applied Mathematics*, 1979, 37(3): 513-538.
- [2] Kariv O, Hakimi S L. An algorithmic approach to network location problems. II: The p -medians. *SIAM Journal on Applied Mathematics*, 1979, 37(3): 539-560.
- [3] Hakimi S L. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 1964, 12(3): 450-459.
- [4] Minieka E. The m -center problem. *SIAM Review*, 1970, 12(1): 138-139.
- [5] Daskin M S. Network and Discrete Location: Models Algorithms and Applications. John Wiley & Sons, 1995.
- [6] Daskin M S. A new approach to solving the vertex p -center problem to optimality: Algorithm and computational results. *Communications of the Operations Research Society of Japan*, 2000, 45(9): 428-436.
- [7] Ilhan T, Pinar M C. An efficient exact algorithm for the vertex p -center problem. http://www.optimization-online.org/DB_HTML/2001/09/376.html, June 2017.
- [8] Elloumi S, Labbé M, Pochet Y. A new formulation and resolution method for the p -center problem. *INFORMS Journal on Computing*, 2004, 16(1): 84-94.
- [9] Al-Khedhairi A, Salhi S. Enhancements to two exact algorithms for solving the vertex p -center problem. *Journal of Mathematical Modelling and Algorithms*, 2005, 4(2): 129-147.
- [10] Hochbaum D S, Shmoys D B. A best possible heuristic for the k -center problem. *Mathematics of Operations Research*, 1985, 10(2): 180-184.
- [11] Martinich J S. A vertex-closing approach to the p -center problem. *Naval Research Logistics*, 1988, 35(2): 185-201.
- [12] Plesník J. A heuristic for the p -center problems in graphs. *Discrete Applied Mathematics*, 1987, 17(3): 263-268.
- [13] Mladenović N, Labbé M, Hansen P. Solving the p -center problem with tabu search and variable neighborhood search. *Networks*, 2003, 42(1): 48-64.
- [14] Hassin R, Levin A, Morad D. Lexicographic local search and the p -center problem. *European Journal of Operational Research*, 2003, 151(2): 265-279.
- [15] Caruso C, Colomi A, Alois L. Dominant, an algorithm for the p -center problem. *European Journal of Operational Research*, 2003, 149(1): 53-64.
- [16] Pacheco J A, Casado S. Solving two location models with few facilities by using a hybrid heuristic: A real health resources case. *Computers & Operations Research*, 2005, 32(12): 3075-3091.
- [17] Davidović T, Ramljak D, Šelmić M, Teodorović D. Bee colony optimization for the p -center problem. *Computers & Operations Research*, 2011, 38(10): 1367-1376.
- [18] Yurtkuran A, Emel E. A modified artificial bee colony algorithm for p -center problems. *The Scientific World Journal*, 2014, 2014: 824196.

- [19] Scaparra MP, Pallottino S, Scutell' M G. Local de grande escala heurísticas de pesquisa para o vértice capacitado p- problema do centro. *Redes*, 2004, 43 (4): 241-255.
- [20] Cheng TCE, Kang LY, Ng C T. Um algoritmo melhorado para o p- problema de centro em gráficos de intervalo com comprimentos de unidade. *Computadores e Pesquisa Operacional*, 2007, 34 (8): 2215-2222.
- [21] Krumke S O. Em uma generalização do p- problema do centro. *Cartas de processamento de informações* 1995, 56 (2): 67-71.
- [22] Arostegui MA Jr, Kadipasaoglu SN, Khumawala B M. Uma comparação empírica de busca tabu, recozimento simulado e algoritmos genéticos para problemas de localização de instalações. *International Journal of Production Economics*, 2006, 103 (2): 742-754.
- [23] Pullan W. Um algoritmo genético memético para o vértice p- problema do centro. *Computação Evolutiva*, 2008, 16 (3): 417-436.
- [24] Laguna M, Marti R. GRASP e reconexão de caminho para minimização de cruzamento de linha reta de 2 camadas. *INFORMS Journal on Computing*, 1999, 11 (1): 44-52.
- [25] Aiex RM, Resende MGC, Pardalos PM, Toraldo G. GRASP com revinculação de caminho para atribuição de três índices. *INFORMS Journal on Computing*, 2005, 17 (2): 224-247.
- [26] Ribeiro CC, Uchoa E, Werneck R F. Um GRASP híbrido com perturbações para o problema de Steiner em grafos. *INFORMS Journal on Computing*, 2002, 14 (3): 228-246.
- [27] Aiex RM, Binato S, Resende MG C. GRASP paralelo com redirecionamento de caminho para agendamento de job shop. *Computação Paralela*, 2003, 29 (4): 393-430.
- [28] Oliveira CAS, Pardalos PM, Resende MG C. GRASP com path-relinking para o problema de atribuição quadrática. *Dentro Proc. o 3º Workshop Int sobre Algoritmos Experimentais e Eficientes*, Maio de 2004, pp.356-368.
- [29] Festa P, PM Pardalos, Resende MGC, Ribeiro C C. Heurísticas randomizadas para o problema de corte máximo. *Métodos e software de otimização*, 2002, 17 (6): 1033-1058.
- [30] Huang WQ, Lv ZP, Shi H. Growth algoritmo para encontrar configurações de baixa energia de proteínas de rede simples. *Revisão Física E*, 2005, 72 (1): 016704.
- [31] Zou P, Zhou Z, Wan YY, Chen GL, Gu J. Nova metaheurística para problemas de otimização combinatória: Escala baseada em interseção. *Journal of Computer Science and Technology*, 2004, 19 (6): 740-751.
- [32] Xu HY, Lv Z, Cheng TC E. Pesquisa local iterada para programação de máquina única com tempos de configuração dependentes da sequência para minimizar o atraso total ponderado. *Journal of Scheduling*, 2014, 17 (3): 271-287.
- [33] Xu HY, Lv ZP, Yin AH, Shen LJ, Buscher U. Um estudo de algoritmos evolutivos híbridos para problemas de programação de máquina única com tempos de configuração dependentes de sequência. *Computadores e Pesquisa Operacional*, 2014, 50: 47-60.
- [34] Glover F. Tabu parte da pesquisa I. *ORSA Journal on Computing*, 1989, 1 (3): 190-206.
- [35] Huang WQ, Zhang DF, Wang H X. Um algoritmo baseado em busca tabu por problemas de satisfação. *Journal of Computer Science and Technology*, 2002, 17 (3): 340-346.
- [36] Lai XJ, Lv Z P. Pesquisa de tabu iterada multistart para problema de coloração de largura de banda. *Computadores e Pesquisa Operacional*, 2013, 40 (5): 1401-1409.
- [37] Wu J, Rosin PL, Sun XF, Martin R R. Melhorando a forma do sombreamento com a pesquisa de tabu interativa. *Journal of Computer Science and Technology*, 2016, 31 (3): 450-462.
- [38] Glover F. Tabu procurar e adaptativo memória programação — Avanços, aplicações e desafios. *Dentro Interfaces em Ciência da Computação e Pesquisa Operacional. Série de Interfaces de Pesquisa Operacional / Ciência da Computação*, Barr RS, Helgason RV, Kennington JL (eds.), Springer 1997, pp.1-75.
- [39] Glover F. Métodos de oscilação estratégica e de partida múltipla - Princípios para explorar a memória adaptativa. *Dentro Ferramentas de computação para otimização de modelagem e simulação de pesquisa de operações / série de interfaces de ciência da computação*, Laguna M, Velarde JLG (eds.), Springer, 2000 pp.1-23.
- [40] Glover F, Laguna M, Mart'ı R. Fundamentos da busca dispersa e reconexão de caminhos. *Controle e cibernética*, 2000, 29 (3): 653-684.
- [41] Peng B, Lv ZP, Cheng TC E. Um algoritmo de revinculação de busca / caminho tabu para resolver o problema de programação da oficina. *Computadores e Pesquisa Operacional*, 2015, 53: 154-164.
- [42] Reinelt G. TSPLIB — Uma biblioteca de problemas do caixeiro viajante. *ORSA Journal on Computing*, 1991, 3 (4): 376-384.
- [43] Floyd R W. Algorithm 97: Shortest path. *Comunicações da ACM*, 1962, 5 (6): Artigo No. 345.
- [44] Lourenço HR, Martin OC, Stützle T. Pesquisa local iterada. *Dentro Manual de metaheurísticas*, Glover F, Kochenberger GA (eds.), Springer, 2003, pp. 320-353.
- [45] Boese K D. Custo versus distância no problema do caixeiro viajante. *Relatório técnico TR-950018 UCLA CS Department*, 1995.
- [46] Stützle T, Dorigo M. Algoritmos ACO para o problema de atribuição quadrática. *Dentro Novas ideias em otimização*, Corne D, Dorigo M, Glover F et al. (eds.), McGraw-Hill Ltd., 1999, páginas 33-50.
- [47] Merz P, Freisleben B. paisagens de fitness, algoritmos meméticos e operadores gananciosos para biparticionamento de grafos. *Computação Evolutiva*, 2000, 8 (1): 61-91.
- [48] Misevicius A. Um algoritmo genético híbrido melhorado: Novos resultados para o problema de atribuição quadrática. *Sistemas Baseados em Conhecimento*, 2004, 17 (2/3/4): 65-73.
- [49] Benlic U, Hao J K. Uma abordagem memética multinível para melhorar o gráfico k- partições. *IEEE Trans. Computação Evolutiva*, 2011, 15 (5): 624-642.
- [50] Geiger M J. Em operadores e topologia de espaço de busca em programação de fluxo de loja multi-objetivo. *European Journal of Operational Research*, 2007, 181 (1): 195-206.



Ai-Hua Yin recebeu seu Ph.D. graduado em ciência da computação e tecnologia pela Universidade de Ciência e Tecnologia de Huazhong, Wuhan, em 2003. Atualmente, ele é pesquisador sênior da Universidade de Finanças e Economia de Jiangxi, Nanchang.

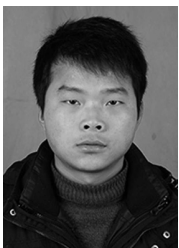
A pesquisa dele interesses incluem agendamento da oficina problema, problema de embalagem retangular e p- problema do centro. Dr. Yin é membro do CCF.



Tao-Qing Zhou recebeu seu diploma de mestre em tecnologia de aplicativos de computador pela South-Central University for Nationalities, Wuhan, em 2003. Atualmente, ele é um candidato a doutorado na Huazhong University of Science and Technology, Wuhan, e um professor sênior na Zhejiang Agriculture and

Universidade Florestal (ZAFU), Hangzhou.

A pesquisa dele os interesses incluem inteligência computacional, solução de problemas de busca combinatória em grande escala, projeto de algoritmos heurísticos para aplicações práticas, problema do caixeiro viajante e problema de roteamento de veículos.



Jun-Wen Ding recebeu seu diploma de BS em ciência da computação da Wuhan Textile University, Wuhan, em 2011, e seu grau de MS em software de computador e teoria da Huazhong University of Science and Technology, Wuhan, em

2014. Atualmente ele está cursando o doutorado. grau com o Laboratório de Inteligência

Computação e Otimização e Escola de Ciência e Tecnologia da Computação, Universidade Huazhong de Ciência e Tecnologia, Wuhan. Seus interesses de pesquisa atuais incluem inteligência artificial, pesquisa operacional e metaheurística para programação de máquinas, programação de job shop e problemas de roteamento de veículos.



Qing-Jie Zhao recebeu seu diploma de bacharelado em ciência da computação pela Universidade de Geociências da China, Wuhan, em 2013, e seu mestrado em tecnologia da computação da Huazhong University of Science and Technology, Wuhan, em 2015. Seus interesses de pesquisa incluem otimização combinatória e evolução

computacional.



Zhi-Peng Lv recebeu seu diploma de bacharelado em matemática aplicada pela Jilin University, Changchun, em 2001, e seu Ph.D. graduado em software de computador e teoria pela Huazhong University of Science and Technology, Wuhan, em 2007. Ele foi bolsista de pós-doutorado

no Laboratoire d'Etude et de Recherche en Informatique d'Angers (LERIA), Departamento de Ciência da Computação, University of Angers, França, de 2007 a 2011. Ele é professor da Escola de Ciência da Computação e Tecnologia da Huazhong University of Science e Tecnologia, Wuhan, e o diretor do Laboratório de Smart Computing e Otimização (SMART), Wuhan. Sua pesquisa está na área de inteligência artificial, inteligência computacional, pesquisa operacional e metaheurísticas adaptativas para resolver problemas de otimização combinatória teórica e de grande escala do mundo real e de satisfação restrita.