Formulações e uma comparação de desempenho para o Problema do Caixeiro Viajante com coleta Seletiva de Prêmios

Guido Pantuza Jr.
Instituto Federal de Minas Gerais
Av. Minas Gerais, 5189, Ouro Verde – Governador Valadares, MG
guido.junior@ifmg.edu.br

Maurício C. de Souza
Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627, Pampulha – Belo Horizonte, MG
mauricio.souza@dep.ufmg.br

Resumo

Este trabalho aborda o Problema do Caixeiro Viajante com coleta Seletiva de Prêmios (PCTSP), o qual é uma generalização do Problema do Caixeiro Viajante (TSP). O PCTSP pertence a um grupo de problemas que envolve dois tipos de decisão: a seleção de quais cidades serão visitadas e a ordem das visitas (*Subtour Problems*). Formulações são propostas e o desempenho é comparado empiricamente através da Relaxação Linear (LP) e do tempo computacional. Os resultados indicam que a formulação PFF-PC, proposta neste trabalho, obteve um valor da LP melhor ou igual do que as demais formulações, para todos os 70 dos 75 problemas-teste em que resolveu a Relaxação Linear no tempo limite de uma hora.
Palavras-chave: Formulações Matemáticas; Desempenho Computacional; PCTSP.

1 Introdução

Este trabalho aborda o Problema do Caixeiro Viajante com coleta Seletiva de Prêmios (*Prize Collecting Traveling Salesman Problem* – PCTSP). Este é uma generalização do Problema do Caixeiro Viajante (*Traveling Salesman Problem* – TSP). Assim como no TSP, o problema é composto por um conjunto de cidades (vértices). A cada cidade está associada uma penalidade e um prêmio. O objetivo é encontrar uma rota S com o menor custo possível. O custo da rota S é composto pela distância percorrida somada à penalidade das cidades não visitadas. Esta rota deve começar e terminar na cidade de origem (vértice raiz r), a qual é fixa. Isto significa que o vértice raiz deve estar presente em todas as soluções viáveis para o problema. Neste problema, cada cidade deve ser visitada, no máximo, uma única vez. Além disso, um prêmio mínimo g deve ser coletado. Note que ao contrário do TSP, nem todos os vértices precisam ser visitados. Para aplicações deste problema veja Balas e Martin (1985), Lopez et al. (1998) e Zhang e Tang (2005).

De uma forma geral, o PCTSP pertence a um grupo de problemas que envolvam dois tipos de decisão: a seleção de quais cidades serão visitadas e qual a ordem das visitas. Este grupo de problemas pode ser chamado de *Subtour Problems*. Alguns dos problemas desta categoria são o *Orienteering Problem* – OP (VOHRA et al., 1987), o *Selective* TSP (LAPORTE; MARTELLO, 1990) e o *Profitable Tour Problem* – PTP (DELL'AMICO et al., 1995). Outros problemas correlatos são o *Quota* TSP (QTSP) e o TSP *with Profits*.

Alguns trabalhos abordam o PCTSP através de algoritmos exatos. Fischetti e Toth (1988) abordam diferentes estruturas presentes no PCTSP para obter *lower bounds*. Através destas estruturas eles utilizam um algoritmo de *Branch and Bound* para resolver o problema. Bérubé et al. (2009) abordam o PCTSP através de um algoritmo de *Branch and Cut*. Os autores aproveitaram-se de alguns

https://proceedings.science/p/85474?lang=pt-br

resultados clássicos do estudo do poliedro do PCTSP (BALAS, 1995), e de extensões de desigualdades válidas para o OP.

Gunawan et al. (2016) apresentam uma extensa e detalhada pesquisa. Eles mostram um abrangente resumo cronológico de diversos trabalhos recentes relacionados aos chamados *Subtour Problems*, em especial o OP. Ainda, eles também abordam o PCTSP e problemas correlatos, como o PTP. Além disso, uma variedade de aplicações recentes e extensões práticas são apresentadas.

Diversos trabalhos comparam as formulações para o TSP Assimétrico (ATSP), tais como Sherali et al. (2006), Oncan et al. (2009), Godinho et al. (2011) e Sarin et al. (2014). Uma vez que, para o PCTSP, nem todos os vértices precisam ser visitados, apesar de ser uma generalização do TSP, a relação de desempenho entre as formulações já estudadas para o ATSP pode ser diferente para o PCTSP. Isto deve-se ao fato de que a estrutura do PCTSP pode ser melhor explorada ao adaptar tais formulações ou através de novas formulações. Este fato já foi observado por Sarin et al. (2014) para o *Multiple Asymmetric TSP* – mATSP. Mesmo assim, não é conhecido um trabalho que trate especificamente de formulações para o PCTSP.

Portanto, o objetivo deste trabalho é adaptar, para o PCTSP, formulações do ATSP já conhecidas da literatura; propor uma formulação que combina características de outras formulações e; compará-las, empiricamente, em termos da Relaxação Linear. Considera-se a versão assimétrica do problema. Ou seja, o custo c_{ij} do percurso entre i e j é diferente de c_{ji}. Caso, $c_{ij} = c_{ji}$ têm-se a versão simétrica.

A seguir, na seção 2, uma fomulação exponencial com eliminação de subrotas é apresentada, seguida de uma pesquisa bibliográfica das principais formulações para o ATSP. Nas seções, 3, 4 e 5 são apresentadas as classificações, bem como as formulações comparadas. Na seção 6 é proposta uma nova formulação válida para o PCTSP. Os resultados são mostrados na seção 7. Finalmente, as conclusões são apresentadas na seção 8.

2 Formulações com eliminação de subrotas

A primeira definição formal para o PCTSP é atribuída à Balas e Martin (1985). Eles adotam uma formulação em um grafo $G = (V, A)$ completo e orientado, no qual V é o conjunto de vértices, sendo $r = 1$ o vértice raiz; A o conjunto de arcos (i, j), tal que i e $j \in V$.

Sejam as variáveis de decisão: x_{ij} = 1 se o arco (i, j) é utilizado na rota e 0 caso contrário; y_i = 1 se vértice i é visitado e 0 caso contrário. E os demais parâmetros: p_i o prêmio coletado no vértice i, caso este seja visitado (incluído na rota); w_i a penalidade incorrida se o vértice i não é visitado; c_{ij} o custo do arco (i, j); e g o prêmio mínimo a ser coletado. O modelo, proposto por Balas e Martin (1985) é representado por (1)-(7):

https://proceedings.science/p/85474?lang=pt-br

$$\min \quad \sum_{i \in V} \sum_{j \in V \setminus \{i\}} c_{ij} x_{ij} + \sum_{i \in V} w_i (1 - y_i) \tag{1}$$

sujeito a:

$$\sum_{j \in V \setminus \{i\}} x_{ij} - y_i = 0 \quad \forall i \in V \tag{2}$$

$$\sum_{i \in V \setminus \{j\}} x_{ij} - y_j = 0 \quad \forall j \in V \tag{3}$$

$$y_r = 1 \tag{4}$$

$$0 \leq y_i \leq 1; \quad x_{ij} \in \{0, 1\} \quad \forall i, j \in V \tag{5}$$

$$\sum_{i \in V - \{j\}} p_i y_i \geq g \tag{6}$$

Não existem subrotas desconexas da raiz r (7)

As restrições de designação, (2) e (3), determinam que se o nó i é visitado, ele será visitado apenas uma única vez. A restrição (4) determina que o vértice raiz r será utilizado. A restrição (6) garante que o prêmio mínimo seja coletado. As restrições (5) definem o domínio das variáveis de decisão. A restrição (7) determina que não existam subrotas desconexas da raiz r.

Um dos principais desafios em criar uma formulação matemática para o PCTSP, e problemas correlatos, está na forma como representar as restrições que eliminam as subrotas (restrição (7)). Estas restrições são conhecidas como SEC's (*Subtour Elimination Constraints*). Diversas formulações com este objetivo foram propostas para o TSP.

Provavelmente, a SEC mais conhecida foi proposta por Dantzig et al. (1954) para o TSP (Restrições (8)):

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \qquad \forall S \subset V : r \in V \setminus S \tag{8}$$

Estas restrições definem facetas para o TSP e, apesar de possuírem um número finito, este é exponencial (BALAS, 1995). Segundo Fischetti e Toth (1988), devido às restrições (2) e (3), a restrição (8), pode ser substituída por (9) (conhecida como *Connectivity Constraint*).

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq y_k \qquad \forall S \subset V : r \in S, k \in V \setminus S \tag{9}$$

Segundo Bérubé et al. (2009), a restrição (9) pode ser reescrita segundo a restrição (10) conhecida como *Generalized Subtour Elimination Constraints* – **GSEC**s.

$$\sum_{i,j \in S} x_{ij} \leq \sum_{i \in S} y_i - y_k \qquad \forall S \subset V : r \in V \setminus S, k \in S \tag{10}$$

Dado que estas restrições são da ordem $\theta(2^n)$, é necessário um Algoritmo de Separação (*Separation Algorithm*) – SA, que identifique as restrições violadas. Dessa forma, elas podem ser gradualmente acrescidas ao modelo à medida que são identificadas.

Entretanto, é possível eliminar as subrotas utilizando um número polinomial de restrições. Neste caso, não é necessário um SA. A primeira formulação polinomial para o ATSP foi proposta por Miller et al. (1960). Ela é conhecida como MTZ. As restrições MTZ são relativamente simples e flexíveis. Logo, elas podem ser facilmente aplicadas a outros problemas que envolvam subrotas. Por isso, Desrochers e Laporte (1991) propuseram uma extensão mais forte que MTZ (para o ATSP),

https://proceedings.science/p/85474?lang=pt-br

denominada DL. Sherali e Driscoll (2002) também se aproveitaram das características de MTZ. Eles propuseram novas SEC's mais fortes que DL (considerando o ATSP), conhecidas como SD.

Gavish e Graves (1978) adotaram outra forma polinomial de representar as SEC's para o ATSP, chamada de GG. Neste caso, as restrições descrevem o fluxo de uma única mercadoria (*commodity*) que percorre os vértices do problema. Por isso, esta classe de formulações receberam o nome de Fluxo de uma Única Mercadoria (*Single Commodity Flow Based Formulations* – SCF) (ONCAN et al., 2009).

Fox et al. (1980) e Gouveia e Manuel (2001) também propuseram novas formulações válidas para o ATSP, respectivamente conhecidas como FGG e GP. FGG aproveita-se da relação entre o ATSP e o Problema de Programação de Máquinas (*Machine Scheduling Problem*). GP são generalizações de MTZ através de restrições que descrevem a ordem de precedência entre os vértices.

As restrições das formulações MTZ, DL, SD, GG, FGG e GP foram omitidas. Isto porque já foi provado que elas possuem uma LP pior do que aquelas apresentadas a seguir. Este fato pode ser observado nos trabalhos de Oncan et al. (2009) e Godinho et al. (2011); e também foi confirmado pelos nossos testes preliminares.

Assim, este trabalho optou por adaptar, para o PCTSP, apenas as formulações que fornecem os melhores valores da Relaxação Linear para o ATSP, segundo os trabalhos recentes de Oncan et al. (2009), Godinho et al. (2011), e Sarin et al. (2014). Estas formulações foram classificadas em grupos baseados nas variáveis utilizadas e são apresentadas a seguir. Ainda, não é de conhecimento dos autores a adaptação de nehuma das formulações apresentadas tanto para o PCTSP quanto para os *Subtour Problems*, exceto por **GSEC**.

3 Formulações de fluxo de múltiplas mercadorias

Em seu trabalho Wong (1980) apresentou uma nova formulação (WONG) para o ATSP. Esta nova formulação utiliza a variável f_{ij}^{kl} que descreve o fluxo de múltiplas mercadorias através dos arcos (i, j) entre os vértices k e l. Ou seja, f_{ij}^{1l} é iqual a 1 apenas se a mercadoria sai do vértice 1 e chega no vértice l passando pelo arco (i, j). Analogamente, f_{ij}^{k1} é iqual a 1 apenas se a mercadoria sai do vértice k e chega ao vértice 1 passando pelo arco (i, j).

Esta formulação proposta para o ATSP é classificada como Formulação de Fluxo de Múltiplas Mercadorias (*Multi-Commodity Flow Formulations* – MCF) (ONCAN et al., 2009). Semelhante à SCF, este grupo de formulações adotam restrições que descrevem o fluxo de mercadorias através dos arcos. Entretanto, este grupo permite múltiplas mercadorias ao invés de uma única, como o SCF.

Claus (1984) generalizou esta formulação, obtendo uma MCF mais compacta para o ATSP. Considerando o ATSP, ela é equivalente a WONG e a DFJ (ONCAN et al., 2009) e, conseguentemente, mais forte que GG e MTZ. Claus (1984) adoutou a variável de fluxo f_{ij}^{k}. Sendo f_{ij}^{k} igual a 1 se o arco (i, j) está na rota entre o vértice k e o vértice origem 1. Dessa forma, ele eliminou metadade das variáveis de mercadorias (*commodities*) e suas respectivas restrições, se comparado com WONG.

A partir da formulação CLAUS é facil generalizar uma MCF para o PCTSP (**CLAUS-PC**). Basta adotar a variável de fluxo f_{ij}^{k} e multiplicar o escalar do lado direto das restrições (RHS) originais por y_k para $k \in V \setminus \{r\}$ e considerar o vértice origem fixo e igual à r. A formulação **CLAUS-PC** é dada por (2)-(6), (11)-(12):

$$\sum_{j \in V | j \neq i} f_{ij}^{k} - \sum_{j \in V | j \neq i} f_{ji}^{k} = \begin{cases} y_k & \text{se } k = i; \\ 0 & \text{caso contrário.} \end{cases} \quad \forall k, i \in V \setminus \{r\}; \tag{11}$$

$$0 \leq f_{ij}^{k} \leq x_{ij} \qquad \forall i, j \in V; k \in V \setminus \{r\} \tag{12}$$

Ainda, se comparado com CLAUS, através dessa generalização (**CLAUS-PC**) foi possível reduzir $(n - 1)$ restrições.

https://proceedings.science/p/85474?lang=pt-br

4 Formulações de caminho e fluxo

Aqui é apresentada uma outra classe de formulações. Esta utiliza variáveis de precedência v_{ij} para representar as SEC's. A variável v_{ij} é contínua, porém posssui uma conotação binária. Ou seja, ela asssume o valor igual a 1 se o vértice i precede o vértice j. Esta relação de precedência não precisa ser direta. Isto significa que o vértice i pode estar em qualquer posição entre o vértice raiz r e o vértice j.

Sherali et al. (2006) apresentaram diversas formulações de tamanho polinomial para o ATSP. Eles as classificaram como Formulações de Caminho e Fluxo *Lifted Path and Flow Formulations* – PFF. Os autores utilizaram a técnica conhecida como *Reformulation-Linearization Technique* (RLT) combinada com as restrições de precedência atavés das variáveis t_{ij}^k e v_{ij}. A variável t_{ij}^k é o artifício utilizado para manter a formulação linear (RLT) substituindo o produto entre $x_{ik}v_{kj}$. Ou seja, $t_{ij}^k = x_{ik}v_{kj}$ para todo $i, j, k \in V \setminus \{r\}$ tal que $i \neq j \neq v$.

Bem como as formulações anteriores, as SEC's de SST podem ser facilmente generalizadas para o PCTSP. Basta alterar o operador lógico para "$\leq$" em (13) e fixar o vértice origem r. Logo, uma nova formulação válida para o PCTSP, **SST-PC** é dada por (2)-(6), (13)-(19):

$$v_{ij} + v_{ji} \leq 1 \qquad \forall i, j \in V \setminus \{r\}; \tag{13}$$

$$(v_{ij} + x_{ji}) + v_{jk} + v_{ki} \leq 2 \qquad \forall i, j, k \in V \setminus \{r\}; \tag{14}$$

$$v_{ij} - x_{ri} \geq 0 \qquad \forall i, j \in V \setminus \{r\}; \tag{15}$$

$$v_{ji} - x_{ir} \geq 0 \qquad \forall i, j \in V \setminus \{r\}; \tag{16}$$

$$0 \leq t_{ij}^k \leq x_{ik} \qquad \forall i, j, k \in V \setminus \{r\}; k \neq j \tag{17}$$

$$\sum_{k \in V \setminus \{r\}; k \neq j} t_{ij}^k + x_{ij} - v_{ij} = 0 \qquad \forall i, j \in V \setminus \{r\}; \tag{18}$$

$$x_{rk} + \sum_{i \in V \setminus \{r\}; i \neq j} t_{ij}^k - v_{kj} = 0 \qquad \forall k, j \in V \setminus \{r\}; \tag{19}$$

5 Formulações de circuito e fluxo

Uma outra forma sofisticada de representar as SEC's foi proposta por Godinho et al. (2011). Este grupo de formulações elimina as subrotas através das variáveis de fluxo $f1_{i,j}^{h,k}$ e $f2_{i,j}^{h,k}$. Estas descrevem o circuito da *commodity* k, na posição h, através dos arcos $(i, j) \in A$. Por isso, este grupo de formulações é classificado como Formulações de Circuito e Fluxo (*Circuit and Flow Based Formulations* – CFF), (GODINHO et al., 2011). Porém, estas variáveis podem ser iterpretadas de outra forma. A variável $f1_{i,j}^{h,k}$ indica se o arco $(i, j) \in A$, na posição h, está no caminho entre o vértice raiz r e o vértice $k \in V \setminus \{r\}$. $f2_{i,j}^{h,k}$ indica se o arco $(i, j) \in A$, na posição h, está no caminho entre o vértice $k \in V \setminus \{r\}$ e o vértice r.

Godinho et al. (2011) propuseram diversas formulações para o *Time-Dependent* TSP, uma variante do TSP clássico. Entre as formulaçãoes propostas, a mais forte é nomeada de GGP. Além das variáveis $f1$ e $f2$, eles também utilizam a variável z_{ij}^h a qual indica se arco $(i, j) \in A$ está na posição h. Segundo os autores, para o ATSP, GGP é mais forte que SST.

A posição h não é relevante no contexto do PCTSP, pois o custo c_{ij} do arco (i, j) não depende da posição. Ainda, na formulação GGP têm-se que $\sum_{h \in V} z_{ij}^h = x_{i,j}$ e $\sum_{h \in V} f1_{i,j}^{h,k} + f2_{i,j}^{h,k} \leq \sum_{h \in V} z_{i,j}^h$ para todo arco $(i, j) \in A$ e $k \in V \setminus \{r\}$. Logo, $\sum_{h \in V} f1_{i,j}^{h,k} + f2_{i,j}^{h,k} \leq x_{i,j}$ para todo arco $(i, j) \in A$ e $k \in V \setminus \{r\}$. Dessa forma, a variável z_{ij}^h pode ser eliminada. Observe que através de (24) o arco (i, j) ou está antes ou depois do vértice k. Ainda, combinando (24) com as restrições de fluxo (21), (22) e (23) o arco (i, j) só será utilizado em uma única posição na solução S. Assim,

https://proceedings.science/p/85474?lang=pt-br

intuitivamente é possível observar que a posição h pode ser omitida. Neste caso, as variáveis $f1^{h,k}_{i,j}$ e $f2^{h,k}_{i,j}$ podem ser substituídas por $f1^{k}_{i,j}$ e $f2^{k}_{i,j}$. Elas possuem a mesma iterpretação, exceto pela posição h omitida. A formulação proposta, **GGP-PC**, é dada por (2)-(6), (20)-(28):

$$\sum_{j \in V \setminus \{r\}} f1^{k}_{r,j} = y_k \qquad \forall k \in V \setminus \{r\} \tag{20}$$

$$\sum_{j \in V \setminus \{r\}} f1^{k}_{i,j} - \sum_{j \in V} f1^{k}_{j,i} = 0 \qquad \forall k, i \in V \setminus \{r\}, k \neq i \tag{21}$$

$$\sum_{j \in V \setminus \{r\}} f2^{k}_{k,j} - \sum_{j \in V} f1^{k}_{j,k} = 0 \qquad \forall k \in V \setminus \{r\} \tag{22}$$

$$\sum_{j \in V \setminus \{r\}} f2^{k}_{i,j} - \sum_{j \in V} f2^{k}_{j,i} = 0 \qquad \forall k, i \in V \setminus \{r\}, k \neq i \tag{23}$$

$$f1^{k}_{i,j} + f2^{k}_{i,j} \leq x_{i,j} \qquad \forall i, j \in V, i \neq j; k \in V \setminus \{r\} \tag{24}$$

$$\sum_{i \in V \setminus \{r\} | i \neq k} f2^{k}_{i,j} = v_{kj} \qquad \forall j, k \in V \setminus \{r\}, j \neq k \tag{25}$$

$$v_{k,j} + v_{j,k} \leq y_j \qquad j, k \in V \setminus \{r\}, j \neq k \tag{26}$$

$$v_{k,j} + v_{j,k} \leq y_k \qquad j, k \in V \setminus \{r\}, j \neq k \tag{27}$$

$$(v_{ij} + x_{ji}) + v_{jk} + v_{ki} \leq 2 \qquad \forall i, j, k \in V \setminus \{r\}; k \neq j \tag{28}$$

6 Nova formulação válida para o PCTSP

Diversas formulações utilizam a variável de precedência v para apertar o modelo. Para mais detalhes, Gouveia e Pesneau (2006) tratam da relação entre v_{ij} e x_{ij} e suas respectivas formulações. Estas variáveis indicam a relação de precedência entre os vértices do problema. Entretanto, a variável t utilizada por Sherali et al. (2006) também pode ser interpretada como uma variável de precedência. Repare que $t^{k}_{ij} = x_{ik}v_{kj}$ para todo $i, j, k \in V \setminus \{r\}$ tal que $i \neq j \neq v$. Ou seja, t^{k}_{ij} indica se o arco (i, k) precede o vértice j, tal que $i \neq k$, $k \neq j$ e $j \neq i$. Ainda, t^{k}_{ik} é igual a 1 se o vértice k não é visitado e i é sucessor direto de r e; t^{k}_{kj} é igual a 1 se o vértice k é sucessor do vértice j na rota entre r e j, ou o vértice j não é visitado. Uma vez que a variável t também descreve relações de precedência, a variável v pode ser redundante, bem como suas respectivas restrições. Note que a variável t^{k}_{ij}, além da relação de precedência, também descreve o fluxo que sai do vértice r e chega ao vértice k. Por isso, esta formulação pode ser classificada como PFF (assim como **SST-PC**). A nova formulação proposta, chamada de **PFF-PC** é dada por (2)-(6), (29)-(32):

$$x_{ir} + x_{ri} \leq y_i \qquad \forall i \in V \setminus \{r\} \tag{29}$$

$$x_{rk} + \sum_{i \in V \setminus \{r,j\}} t^{k}_{ij} - y_k = 0 \qquad \forall k, j \in V \setminus \{r\}, k \neq j; \tag{30}$$

$$\sum_{k \in V \setminus \{r\}} t^{k}_{ij} + x_{ij} - y_i = 0 \qquad \forall i, j \in V \setminus \{r\}, i \neq j; \tag{31}$$

$$0 \leq t^{k}_{ij} \leq x_{ik} \qquad \forall i, j, k \in V \setminus \{r\}, k \neq j, k \neq i, i \neq j; \tag{32}$$

A restrição (30) garante que se o vértice k está ativo (ou seja, $y_k = 1$), ou ele está diretamente conectado à raiz (x_{rk}) ou está em algum lugar do caminho entre i e j. A restrição (31) garante que se o vértice i está ativo, ou ele está diretamente conectado ao vértice j ou a outro vértice k.

https://proceedings.science/p/85474?lang=pt-br

Dado que para qualquer solução viável do PCTSP existe um vértice j tal que $x_{jr} = 1$, restrições (3) e (4). As restrições (30), (31) e (32) garantem o fluxo contínuo no caminho entre os vértices até r, através dos arcos ativos.

Assume-se que o prêmio mínimo g é um valor percentual do somatório do prêmio para todo vértice $i \in V$. Neste caso, para instâncias suficientemente grandes, considera-se que a solução ótima S^* possui mais do que três vértices ($|S^*| > 3$) e contém o vértice r, restrição (4). Repare que as restrições de fluxo (30) e (31) não consideram o vértice r. Porém, uma solução viável necessita de pelo menos três vértices (o arco (i, k) predecessor de j), proibindo, portanto, subrotas menores que 3 ($|S| < 3$). Note que (29) proíbem subrotas que contenham r e sejam menores ou iguais a 2 ($|S| \leq 2$); e se combinada com (2)-(3), garantem que r esteja conectado diretamente a um vértice predecessor e um sucessor. As restrições (2)-(4) e (29)-(31), combinadas, proíbem, simultaneamente, soluções desconexas da raiz e subrotas com ($|S| < 3$). Ou seja, uma solução viável possui pelo menos três vértices mais r. Uma vez que (13)-(16) (**SST-PC**) são responsáveis por inibir subrotas menores que 3 e a conectividade com r, estas restrições, bem como a variável v tornam-se redundantes (ou dominadas).

7 Experimentos computacionais

Dado que a formulação **GSEC** possui um número de restrições exponenciais, utilizou-se um Algoritmo de Separação para identificar as restrições violadas. Este algoritmo pode ser resolvido em tempo polinomial. Para tanto, foi utilizado um algoritmo de fluxo-máximo para verificar se existe algum corte mínimo ($\delta(S)$) entre os vértices r e i, tal que $y_i > 0$, na solução linearmente relaxada que viole a restrição $\delta(S) \geq 2$. O Algoritmo de Separação implementado adota o algoritmo de fluxo-máximo baseado na implementação proposta por Dinitz (2006), a qual possui uma complexidade $\theta(n^2m)$, sendo n o número de vértices e m o número de arcos.

Para comparar o desempenho entre as formulações, os experimentos foram realizados com o GUROBI versão 7.5.2, 64 bits, através de um PC Intel® Core™ i7-2600 CPU 3.40GHz e 8 GB de RAM sob plataform a Linux, 64 bits.

Os resultados foram obtidos através de duas Classes de instâncias (A, B). A primeira, retirada de Chaves e Lorena (2008), nomeada como A, é composta por 9 problemas-teste diferentes. Estes problemas foram gerados através dos mesmos critérios que Fischetti e Toth (1988) e $g = 75\%$ do somatório de todos os prêmios associados às cidades. A Classe B foi foi adaptada a partir da TSPLIB, Reinelt (1991), inicialmente proposta para o TSP, e possui 66 problemas-teste. Neste caso, a matriz de custos das viagens entre dois vértices foi mantida a mesma. Para obter os demais parâmetros, adotou-se os mesmos critérios que Dell'Amico et al. (1995) e Bérubé et al. (2009). Esta classe de problemas-teste foi divida em três subclasses, de acordo com o valor de g. A saber: B-2 para $g = 20\%$, B-5 para $g = 50\%$ e B-8 para $g = 80\%$.

7.1 Resultados

Considerou-se como parâmetros de desempenho o gap e o melhor tempo de execução (CPU_{LP}). O gap, é calculado segundo a Equação $gap = 100 \times \frac{(OPT-LP)}{OPT}$. Na qual, OPT é o valor da solução ótima e LP o valor da solução encontrada através da Relaxação Linear. A solução ótima foi obitida através de **GSEC**, a qual foi a única a resolver todos os problemas até a otimalidade (domínio das variáveis inteiro).

A Tabela 1 mostra o número de problemas Linearmente Relaxados ($\#SOLVED_{LP}$), resolvidos por cada formulação em até uma hora.

A Tabela 2 apresenta os resultados das instâncias nas quais todas as formulações foram capazes de resolver a LP e encontraram o mesmo gap, exceto por **GSEC**. Isto porque **GSEC** teve um gap pior do que as demais, para todos as intâncias. Uma vez que o gap é igual para todos, estes foram

https://proceedings.science/p/85474?lang=pt-br

Tabela 1 – Número de problemas Linearmente Relaxados, resolvidos por cada formulação, em até uma hora (3600 segundos).

PAR.	CLAUS-PC	GPP-PC	GSEC	PFF-PC	SST-PC
$\#SOLVED_{LP}$	74	71	75	70	69

omitidos. Assim, na primeira linha é mostrado o gap médio ($AVG.gap$), seguido pelo tempo médio de execução ($AVG.CPU_{LP}$). Nas demais linhas são mostrados o menor ($MIN.gap$) e o maior gap ($MAX.gap$) para os problemas-teste resolvidos.

Tabela 2 – Resultados da comparação do desempenho: gap e tempo de execução CPU_{LP}, em segundos, para os problemas-teste nos quais todas as formulações resolveram a LP.

PAR.	CLAUS-PC	GPP-PC	GSEC	PFF-PC	SST-PC
$AVG.gap$	1,2	1,2	9,4	1,2	1,2
$AVG.CPU_{LP}$	25	474	0	70	530
$MIN.gap$	0,0	0,0	1,8	0,0	0,0
$MAX.gap$	9,1	9,1	38,8	9,1	9,1

A Tabela 3 apresenta os resultados das instâncias nas quais todas as formulações foram capazes de resolver a LP e encontraram um gap diferente. Assim como na Tabela 2, na primeira linha é mostrado o gap médio ($AVG.gap$), seguido pelo tempo médio de execução ($AVG.CPU_{LP}$).

Tabela 3 – Comparação do desempenho através do gap e do tempo médio de execução CPU_{LP}, em segundos.

CLASSE	PROB.	PAR.	CLAUS-PC	GPP-PC	GSEC	PFF-PC	SST-PC
		$AVG.gap$	6,7	6,7	23,7	**5,0**	6,1
		$AVG.CPU_{LP}$	63	1047	10	288	1062
B-2	ft70	gap	1,6	1,6	3,9	**1,4**	1,6
		CPU_{LP}	23	35	26	5	40
	hk48	gap	14,7	14,7	27,9	**12,0**	13,8
		CPU_{LP}	27	143	3	93	143
	pr76	gap	14,4	14,4	45,1	**8,0**	11,8
		CPU_{LP}	176	3407	17	890	3558
	ft53	gap	0,7	0,7	4,9	**0,6**	**0,6**
		CPU_{LP}	17	38	0	18	16
	ry48p	gap	10,6	10,6	19,6	**8,7**	9,9
		CPU_{LP}	19	38	0	35	47
B-5	hk48	gap	2,1	2,1	24,8	**1,5**	1,9
		CPU_{LP}	24	138	3	81	145
	pr76	gap	3,1	3,1	39,7	**2,6**	2,8
		CPU_{LP}	157	3529	17	892	3484

Finalmente, a Tabela 4 apresenta os demais resultados para os problemas-teste, nos quais pelo menos uma formulação conseguiu resolver o problema linearmente relaxado. O símbolo "**–**" indica que a formulação não foi capaz de resolver o problema em até uma hora (3600 seg.). O símbolo "******" mostra os casos nos quais as formulações não resolveram a LP devido à falta de memória.

Repare que **GSEC**, seguida por **CLAUS-PC** resolveram o maior número de instâncias e, **SST-PC**, o menor (Tab. 1). Além disso, **GSEC** obteve um tempo computacional muito superior às demais. Apesar de ser a mais rápida para todos os problemas, **GSEC** teve o pior gap para todos os problemas-teste (Tab. 2 e 3). Ainda, esta foi a formulação que resolveu o maior número de problemas-teste Linearmente Relaxados; e a única que não apresentou falta de memória. As demais formulações apresentaram falta de memória computacional para problemas com $n > 100$ (Tab. 4). Exceto por **GSEC**, todas as formulações testadas tiveram dificuldades para resolver problemas com $n \geq 100$

Através dos resultados é possível observar que a formulação **PFF-PC** obteve o melhor gap médio ($AVG.gap$) seguida por **SST-PC** e **GSEC** o pior (Tab. 2 e 3). Ainda, a formulação **PFF-PC** foi dominante, (ou mais forte ou mais apertada), em relação às demais. Ou seja, apresentou um gap melhor ou igual todos os 70 dos 75 problemas-teste em que resolveu a relaxação linear no tempo

https://proceedings.science/p/85474?lang=pt-br

Tabela 4 – Resultados da comparação do desempenho: gap e tempo médio de execução CPU_{LP}, em segundos, para os demais problemas-teste

CLASSE	PROB	PAR.	CLAUS-PC	GPP-PC	GSEC	PFF-PC	SST-PC
A	v100a	gap	–	**	2,5	–	**
		CPU_{LP}	–	**	2	–	**
	v100b	gap	0,1	**	2,9	–	**
		CPU_{LP}	546	**	1	–	**
B-2	kro100	gap	3,1	**	24,4	–	–
		CPU_{LP}	1200	**	25	–	–
B-5	kro100	gap	1,5	1,5	23,4	–	–
		CPU_{LP}	856	3365	33	–	–
B-8	kro100	gap	1,0	**	12,4	**	**
		CPU_{LP}	1067	**	44	**	**
	pr76	gap	3,3	3,3	16,5	3,3	–
		CPU_{LP}	138	3413	17	445	–

limite de uma hora. Este fato indica que esta formulação, para o PCTSP, pode ser mais forte que as demais. **SST-PC**, se comparada com **CLAUS-PC** e **GPP-PC**, apresentou um gap melhor ou igual para todos os problemas. Logo, ela pode ser dominante em relação à **CLAUS-PC** e **GPP-PC**. As formulações **CLAUS-PC** e **GPP-PC** obtiveram o mesmo gap para todos as intâncias testadas (Tab. 2 e 3). Isto é um indício de que elas podem ser equivalentes.

8 Conclusão

Este trabalho compara, empiricamente, 5 formulações para o PCTSP Assimétrico. Estas formulações são novas (**PFF-PC**) ou foram adaptadas a partir do ATSP (**CLAUS-PC**, **SST-PC**, **GGP-PC**), para o PCTSP pela primeira vez, além de **GSEC** já utilizada por Bérubé et al. (2009). O desempenho dessas formulações são avaliadas a partir do valor da Relaxação linear (LP) e do tempo médio de execução usando o GUROBI 7.5.2, 64 bits.

Através dos resultados foi possível observar que a formulação **PFF-PC**, proposta neste trabalho, foi dominante em relação às demais. Conforme observado por Sarin et al. (2014), considerando o mesmo grupo de formulações, aquela que foi proposta diretamente para o PCTSP obteve melhor resultado do que as adaptações a partir do ATSP.

A memória computacional e o tempo demandados, pelas formulações polinomiais, mostraram-se o maior desafio para a utilização das mesmas. Assim, o uso de outros algoritmos exatos, como um *Branch-and-Bound* baseado na Relaxação Lagrangena ao invés da Relaxação Linear, combinado com planos de corte, podem viabilizar a aplicação destas formulações para problemas mais complexos.

Referências

BALAS, E. The prize collecting traveling salesman problem: II polyhedral results. *Networks*, v. 25, n. 4, p. 199–216, 1995.

BALAS, E.; MARTIN, G. *ROLL-A-ROUND: software package for scheduling the rounds of a rolling mill*. 104, Maples Heights Road, Pittsburgh, 1985.

BÉRUBÉ, J.-F.; GENDREAU, M.; POTVIN, J.-Y. A branch-and-cut algorithm for the undirected prize collecting traveling salesman problem. *Networks*, v. 54, n. 1, p. 56–67, 2009.

CHAVES, A. A.; LORENA, L. A. N. Hybrid metaheuristic for the prize collecting travelling salesman problem. In: HEMERT, J. V.; COTTA, C. (Ed.). *Evolutionary Computation in Combinatorial Optimization*. Berlin: Springer, 2008. p. 123–134.

CLAUS, A. A new formulation for the travelling salesman problem. *SIAM Journal on Algebraic Discrete Methods*, v. 5, n. 1, p. 21–25, 1984.

https://proceedings.science/p/85474?lang=pt-br

DANTZIG, G. B.; FULKERSON, D. R.; JOHNSON, S. M. Solution of a large scale traveling salesman problem. *Operational Research*, v. 2, n. 4, p. 393–410, 1954.

DELL'AMICO, M.; MAFFIOLI, F.; VARBRAND, P. On prize-collecting tours and the asymmetric travelling salesman problem. *International Transactions in Operational Research*, v. 2, n. 3, p. 297–308, 1995.

DESROCHERS, M.; LAPORTE, G. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, v. 10, n. 1, p. 27–36, 1991.

DINITZ, Y. Dinitz' algorithm: The original version and even's version. In: GOLDREICH, O.; ROSENBERG, A. L.; SELMAN, A. L. (Ed.). *Theoretical Computer Science: Essays in Memory of Shimon Even*. Berlin, Heidelberg: Springer, 2006. p. 218–240.

FISCHETTI, M.; TOTH, P. An additive approach for the optimal solution of the prize-colleting travelling salesman problem. In: *Vehicle Routing: Methods and Studies*. Holland: Elsevier Science, 1988. p. 319–343.

FOX, K. R.; GAVISH, B.; GRAVES, S. C. Technical note-an n-constraint formulation of the time-dependent traveling salesman problem. *Operations Research*, v. 28, n. 4, p. 1018–1021, 1980.

GAVISH, B.; GRAVES, S. C. *The travelling salesman problem and related problems*. Operations Research Center, Massachusetts Institute of Technology, Working Paper GR-078-78. Cambridge, 1978.

GODINHO, M. T.; GOUVEIA, L.; PESNEAU, P. On a time-dependent flow-based formulation and an updated classification of formulations for the travelling salesman problem. In: *Progress in Combinatorial Optimization*. Istambul, Turkey: Willey, 2011. p. 223–254.

GOUVEIA, L.; MANUEL, P. J. The asymmetric travelling salesman problem: on generalizations of disaggregated Miller-Tucker-Zemlin constraints. *Discrete Applied Mathematics*, v. 112, n. 1, p. 129 – 145, 2001.

GOUVEIA, L.; PESNEAU, P. On extended formulations for the precedence constrained asymmetric traveling salesman problem. *Networks*, v. 48, n. 2, p. 77–89, 2006.

GUNAWAN, A.; LAU, H. C.; VANSTEENWEGEN, P. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, v. 255, n. 2, p. 315–332, 2016.

LAPORTE, G.; MARTELLO, S. The selective travelling salesman problem. *Discrete Applied Mathematics*, v. 26, n. 2, p. 193–207, 1990.

LOPEZ, L.; CARTER, M. W.; GENDREAU, M. The hot strip mill production scheduling problem: A tabu search approach. *European Journal of Operational Research*, v. 106, n. 2-3, p. 317–335, 1998.

MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. *Journal of the Association for Computing Machinery*, v. 7, n. 4, p. 326–329, 1960.

ONCAN, T.; ALTINEL, I. K.; LAPORTE, G. A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers and Operations Research*, v. 36, n. 3, p. 637–654, 2009.

REINELT, G. Tsplib – a traveling salesman problem library. *ORSA Journal on Computing*, v. 3, n. 4, p. 376–384, 1991.

https://proceedings.science/p/85474?lang=pt-br

SARIN, S. C.; SHERALI, H. D.; JUDD, J. D.; TSAI, P. J. Multiple asymmetric traveling salesmen problem with and without precedence constraints: Performance comparison of alternative formulations. *Computers and Operations Research*, v. 51, p. 64–89, 2014.

SHERALI, H. D.; DRISCOLL, P. J. On tightening the relaxations of Miller-Tucker-Zemlin formulations for asymmetric traveling salesman problems. *Operations Research*, v. 50, n. 4, p. 656–669, 2002.

SHERALI, H. D.; SARIN, S. C.; TSAI, P.-F. A class of lifted path and flow-based formulations for the asymmetric traveling salesman problem with and without precedence constraints. *Discrete Optimization*, v. 3, n. 1, p. 20–32, 2006.

VOHRA, R.; LEVY, L.; GOLDEN, B. L. The orienteering problem. *Naval Research Logistics*, v. 34, n. 3, p. 307–318, 1987.

WONG, R. T. Integer programming formulations of the traveling salesman problems. In: *IEEE international conference of circuits and computers – (ICCC80)*. New York, USA: , 1980. p. 149–152.

ZHANG, Y.; TANG, L. Color coated coil scheduling in cold rolling plant using artificial neural network. In: *2005 ICSC Congress on Computational Intelligence Methods and Applications*. Istambul, Turkey: , 2005.

https://proceedings.science/p/85474?lang=pt-br