

NGHIÊN CỨU ỨNG DỤNG THUẬT TOÁN ACO (ANT COLONY OPTIMIZATION) TỐI ƯU THỜI GIAN VÀ CHI PHÍ CHO DỰ ÁN XÂY DỰNG

Phạm Hồng Luân⁽¹⁾, Dương Thành Nhân⁽²⁾

(1) Trường Đại học Bách Khoa, ĐHQG-HCM

(2) Công ty CP Tài trợ và Phát triển địa ốc R.C

(Bài nhận ngày 30 tháng 09 năm 2009, hoàn chỉnh sửa chữa ngày 24 tháng 12 năm 2009)

TÓM TẮT: Bài toán tối ưu thời gian - chi phí là một trong những khía cạnh quan trọng nhất của quản lý dự án xây dựng. Để cực đại hóa lợi nhuận, các nhà lập kế hoạch xây dựng phải cố gắng tìm cách tối ưu đồng thời thời gian và chi phí. Trong nhiều năm qua, nhiều nghiên cứu đã được thực hiện nhằm nghiên cứu mối quan hệ thời gian - chi phí, các kỹ thuật được ứng dụng từ phương pháp tìm kiếm, phương pháp toán học cho đến thuật giải di truyền. Trong bài báo này, một thuật toán tối ưu dựa trên nền tảng của sự tiến hóa, với tên gọi tối ưu đàn kiến (ACO) được ứng dụng để giải quyết bài toán tối ưu đa mục tiêu thời gian - chi phí. Bằng cách kết hợp với phương pháp trọng số thích ứng sửa đổi (MAWA), mô hình sẽ tìm ra các lời giải tối ưu. Mô hình ACO-TCO sẽ được phát triển bằng một chương trình máy tính trên nền Visual Basic. Một ví dụ sẽ được phân tích để minh họa khả năng của mô hình cũng như so sánh với các phương pháp trước đây. Kết quả chỉ ra rằng phương pháp này có khả năng tìm ra những kết quả tốt hơn mà không cần sử dụng quá nhiều đến máy điện toán, từ đó cung cấp một phương tiện hữu hiệu để hỗ trợ các nhà lập kế hoạch và quản lý trong việc lựa chọn những quyết định về thời gian - chi phí một cách hiệu quả.

Từ khóa: Ant colony optimization (ACO), genetic algorithm, GA, MAWA, ACO-TCO.

1. GIỚI THIỆU

Với sự ra đời của các sáng kiến cũng như các kỹ thuật xây dựng hiệu quả, các sáng kiến trong quản lý và các phương pháp phân phát, thời gian xây dựng đã được cải thiện một cách rõ rệt trong vòng vài thập kỷ gần đây. Trên quan điểm của chủ đầu tư, một dự án kết thúc sớm sẽ giúp giảm bớt khoản nợ về tài chính và cho phép họ thu lại nguồn vốn đầu tư sớm hơn. Mặt khác, các nhà thầu sẽ tiết kiệm được chi phí gián tiếp và giảm thiểu được nguy cơ lạm phát cũng như số lượng nhân công nếu thời gian của dự án có thể được rút ngắn. Trên cơ sở này, các nhà lập kế hoạch và quản lý dự án đều cố gắng bảo đảm rằng tất cả các hoạt động xây

dựng đều phải hoàn thành không những đúng thời gian tiến độ mà phải vượt tiến độ đề ra.

Bài toán tối ưu thời gian - chi phí (time-cost optimization - TCO) là một trong những bài toán quan trọng nhất của việc lập và quản lý dự án. Các nhà quản lý dự án phải lựa chọn những nguồn tài nguyên thích hợp, bao gồm: kích cỡ tổ đội, vật tư thiết bị, máy móc... cũng như phương pháp và kỹ thuật thi công để thực hiện các công tác của dự án. Nói chung, có một mối quan hệ tương quan giữa thời gian và chi phí để hoàn thành một công tác; chi phí thấp thì thời gian thực hiện công tác sẽ kéo dài, và ngược lại. Những bài toán loại này thường rất khó giải quyết bởi vì chúng không có một đáp

án duy nhất. Vì vậy, nhiệm vụ của các nhà quản lý dự án là phải xem xét, đánh giá một cách kỹ lưỡng nhiều phương pháp khác nhau nhằm đạt được một kết quả cân bằng tối ưu giữa thời gian và chi phí.

Các phương pháp để giải quyết bài toán TCO hiện tại có thể được chia thành ba nhóm: phương pháp tìm kiếm (heuristic methods), phương pháp quy hoạch toán học (mathematical programming models) và các thuật toán tối ưu dựa trên nền tảng của sự tiến hóa (evolutionary-based optimization algorithms_EOAs). Phương pháp tìm kiếm là một kỹ thuật tìm kiếm dựa trên ý kiến chủ quan của của người ra quyết định. Các phương pháp tìm kiếm tiêu biểu dùng để giải quyết bài toán TCO gồm : phương pháp Fondahl (1963), phương pháp khung (Prager 1963), phương pháp độ dốc chi phí hiệu quả (Siemens 1971),... Phương pháp quy hoạch toán học sử dụng các chương trình toán học như quy hoạch tuyến tính (linear programming_LP), được giới thiệu bởi Kelly (1961), Hendrickson and Au (1989) và Pagnoni (1990) để mô hình hóa mối quan hệ tuyến tính giữa thời gian – chi phí. Ngoài ra, quy hoạch số nguyên (integer programming_IP) được giới thiệu bởi Meyer & Shaffer (1963) để giải quyết cả mối quan hệ tuyến tính và rời rạc giữa thời gian – chi phí. Gần đây, Burns cùng các cộng sự (1996) đã phát triển một mô hình lai ghép LP/IP nhằm thiết lập đáp án chính xác cho bất kỳ khoảng thời gian mong muốn nào.

Cả hai phương pháp tìm kiếm và quy hoạch toán học đều có những điểm mạnh cũng

như nhược điểm riêng trong việc giải quyết bài toán TCO. Tuy nhiên, đối với các dự án lớn với sơ đồ mạng lớn, thì cả phương pháp tìm kiếm cũng như phương pháp quy hoạch toán học đều không thể đạt được lời giải tối ưu một cách hiệu quả. Với mục tiêu đạt được lời giải tối ưu cho bài toán TCO, nhiều nhà nghiên cứu đã bắt đầu khám phá khả năng sử dụng các phương pháp tiên tiến, như là EOAs. EOAs (evolutionary-based optimization algorithms) là phương pháp nghiên cứu dựa trên việc mô phỏng quá trình tiến hoá của thế giới tự nhiên hoặc hành vi xã hội của các loài. Trong số các EOAs, GAs (genetic algorithms) - thuật giải di truyền - được sử dụng rộng rãi nhất nhằm thu được lời giải tối ưu cho các bài toán tối ưu đa mục tiêu trong nhiều lĩnh. Chẳng hạn, Feng và các cộng sự (1997) đã phát triển một mô hình GA mà về cơ bản là sự cải thiện mô hình lai ghép được phát minh bởi Liu và các cộng sự (1995). Feng và các cộng sự (2000) phát triển một mô hình GA cho bài toán cân bằng thời gian-chi phí trong xây dựng. Bên cạnh thuật giải di truyền, nhiều kỹ thuật EOA khác lấy cảm hứng từ nhiều tiến trình khác nhau trong tự nhiên cũng đã được phát triển như thuật toán memetic (Moscato 1989), tối ưu bầy đàn (Kenedy và Eberhart 1995)...

Vào đầu thập niên 90, một thuật toán với tên gọi *Tối ưu đàn kiến* (Ant Colony Optimization_ACO) được đề xuất như là một phương pháp mới trong việc tìm kiếm lời giải tối ưu cho những bài toán tối ưu đa mục tiêu. ACO lần tiên được ứng dụng để giải quyết bài toán người thương gia TSP (Traveling

Salesmen Problem), và gần đây nó đã được mở rộng và cải tiến để áp dụng cho nhiều bài toán tối ưu khác nhau.

Bài báo này sẽ đi sâu nghiên cứu và ứng dụng thuật toán ACO - là một phương pháp tìm kiếm nên cũng là một dạng heuristic - để giải quyết bài toán tối ưu đa mục tiêu TCO trong một dự án xây dựng. Việc phát triển một chương trình máy tính dựa trên mô hình thuật toán được nghiên cứu, nhằm kiểm tra kết quả dựa trên số liệu của một dự án xây dựng thực tế, cũng như so sánh với những phương pháp trước đây, cũng sẽ được xem xét trong bài báo này.

2. THUẬT TOÁN ACO

ACO (*Ant Colony Optimization – Tối ưu đàn kiến*) là một phương pháp nghiên cứu lấy cảm hứng từ việc mô phỏng hành vi của đàn kiến trong tự nhiên nhằm mục tiêu giải quyết các bài toán tối ưu phức tạp.

Được giới thiệu lần đầu tiên vào năm 1991 bởi A. Colormi và M. Dorigo, *Giải thuật kiến* đã nhận được sự chú ý rộng lớn nhờ vào khả năng tối ưu của nó trong nhiều lĩnh vực khác nhau. Khái niệm ACO lấy cảm hứng từ việc quan sát hành vi của đàn kiến trong quá trình chúng tìm kiếm nguồn thức ăn. Người ta đã khám phá ra rằng, đàn kiến luôn tìm được đường đi ngắn nhất từ tổ của chúng đến nguồn thức ăn. Phương tiện truyền đạt tín hiệu được kiến sử dụng để thông báo cho những con khác trong việc tìm đường đi hiệu quả nhất chính là mùi của chúng (*pheromone*). Kiến để lại vết mùi trên mặt đất khi chúng di chuyển với mục

đích đánh dấu đường đi cho các con theo sau. Vết mùi này sẽ bay hơi dần và mất đi theo thời gian, nhưng nó cũng có thể được củng cố nếu những con kiến khác tiếp tục đi trên con đường đó lần nữa. Dần dần, các con kiến theo sau sẽ lựa chọn đường đi với lượng mùi dày đặc hơn, và chúng sẽ làm gia tăng hơn nữa nồng độ mùi trên những đường đi được yêu thích hơn. Các đường đi với nồng độ mùi ít hơn rốt cuộc sẽ bị loại bỏ và cuối cùng, tất cả đàn kiến sẽ cùng kéo về một đường đi mà có khuynh hướng trở thành đường đi ngắn nhất từ tổ đến nguồn thức ăn của chúng (Dorigo và Gambardella 1996).

Để bắt chước hành vi của các con kiến thực, Dorigo xây dựng các con kiến nhân tạo (*artificial ants*) cũng có đặc trưng sản sinh ra vết mùi để lại trên đường đi và khả năng lần vết theo nồng độ mùi để lựa chọn con đường có nồng độ mùi cao hơn để đi. Gắn với mỗi cạnh (i,j) nồng độ vết mùi τ_{ij} và thông số heuristic η_{ij} trên cạnh đó.

Ban đầu, nồng độ mùi trên mỗi cạnh (i,j) được khởi tạo bằng một hằng số c , hoặc được xác định theo công thức :

$$\tau_{ij} = \tau_0 = \frac{m}{C^m}, \forall (i,j) \quad (1)$$

Trong đó :

- τ_{ij} : nồng độ vết mùi trên cạnh i,j
- m : số lượng kiến
- C^m : chiều dài hành trình cho bởi phương pháp tìm kiếm gần nhất.

Tại đỉnh i , một con kiến k sẽ chọn đỉnh j chưa được đi qua trong tập láng giềng của i

theo một quy luật phân bố xác suất được xác định theo công thức sau:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, j \in N_i^k \quad (2)$$

Trong đó :

- p_{ij}^k : xác suất con kiến k lựa chọn cạnh i, j
- α : hệ số điều chỉnh ảnh hưởng của τ_{ij}
- η_{ij} : thông tin heuristic giúp đánh giá

chính xác sự lựa chọn của con kiến khi quyết định đi từ đỉnh i qua đỉnh j ; được xác định theo công thức :

$$\eta_{ij} = 1/d_{ij} \quad (3)$$

- d_{ij} : khoảng cách giữa đỉnh i và đỉnh j
- β : hệ số điều chỉnh ảnh hưởng của η_{ij}
- N_i^k : tập các đỉnh láng giềng của

i mà con kiến k chưa đi qua

Quy luật này mô phỏng hoạt động của một vòng quay xỏ số nên được gọi là kỹ thuật bánh xe xỏ số.

Cho một hằng số $0 \leq q_0 \leq 1$ và một số $0 \leq q \leq 1$ được tạo ra một cách ngẫu nhiên. Con kiến k ở đỉnh i sẽ lựa chọn đỉnh j kế tiếp để đi theo một quy tắc lựa chọn được mô tả bởi công thức sau :

$$j = \begin{cases} \arg \max_{l \in N_i^k} [(\tau_{il})^\alpha \times (\eta_{il})^\beta] & \text{Nếu } q \leq q_0 \\ J & \text{Ngược lại} \end{cases} \quad (4)$$

Trong đó :

▪ q : giá trị được lựa chọn một cách ngẫu nhiên với một xác suất không thay đổi trong khoảng $[0,1]$.

▪ $0 \leq q_0 \leq 1$: là một hằng số cho trước.

▪ J : là một biến số ngẫu nhiên được lựa chọn theo sự phân bố xác suất cho bởi quy luật phân bố xác suất theo công thức (2).

Sau khi cũng như trong quá trình các con kiến tìm đường đi, các vết mùi (τ_{ij}) trên mỗi cạnh sẽ được cập nhật lại, vì chúng bị biến đổi do quá trình bay hơi cũng như quá trình tích lũy mùi khi các con kiến đi trên cạnh đó.

Sau mỗi vòng lặp, vết mùi trên mỗi cạnh được cập nhật lại theo công thức sau:

$$\tau_{ij}(t+1) = (1-\rho) \times \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad \forall (i,j) \quad (5)$$

Trong đó :

▪ $0 < \rho \leq 1$: tỷ lệ bay hơi của vết mùi.

▪ $\Delta \tau_{ij}^k(t)$: lượng mùi mà con kiến k để

lại trên cạnh ij , được xác định như sau :

$$\Delta \tau_{i,j}^k = \begin{cases} \frac{Q}{f(k)} & \text{Nếu con kiến } k \text{ đi qua cạnh } (i,j) \\ 0 & \text{Ngược lại} \end{cases} \quad (6)$$

▪ Q : là một hằng số.

▪ $f(k)$: giá trị mục tiêu trong mỗi vòng lặp.

3. TỐI ƯU ĐA MỤC TIÊU (MULTI-OBJECTIVE OPTIMIZATION)

Bài toán TCO là một bài toán tối ưu đa mục tiêu. Không giống như những bài toán tối ưu đơn mục tiêu mà lời giải tối ưu tồn tại một

cách rõ ràng, tối ưu đa mục tiêu thích hợp đối với những bài toán có hơn một mục tiêu đối lập. Bài toán TCO được mô tả bởi một chuỗi lời giải mà ta không dễ so sánh, và rất khó nếu không muốn nói là không thể thu được lời giải tốt nhất một cách rõ ràng cho tất cả các mục tiêu. Mỗi một hàm mục tiêu có thể đạt được điều kiện tối ưu của nó tại những điểm khác nhau nhờ vào sự thiếu hụt tiêu chuẩn thống nhất về sự tối ưu. Do đó, các nhà lập kế hoạch và quản lý phải áp dụng sự đánh giá về mặt kỹ thuật của họ trong việc lựa chọn đáp án tốt nhất thu được từ một bộ các đáp án tối ưu dọc theo biên Pareto (Zheng và các cộng sự 2005).

Được trình bày bởi Vilfredo Pareto vào thế kỷ 19, khái niệm tối ưu Pareto là một công cụ được chấp nhận trong việc so sánh giữa hai đáp án trong bài toán tối ưu đa mục tiêu mà không có tiêu chuẩn thống nhất về sự tối ưu. Các đáp án như thế đòi hỏi phải hy sinh ít nhất một mục tiêu khác khi muốn cải thiện bất kỳ một mục tiêu nào (Gen và Cheng 2000). Một đáp án (x^*) không bị trội bởi một đáp án (x) khác nếu nó có ít nhất một tiêu chuẩn tốt hơn khi so sánh với (x). Vùng được định nghĩa bởi tối ưu Pareto được gọi là biên Pareto (Pareto front), và mục tiêu của tối ưu đa mục tiêu là thiết lập toàn bộ biên Pareto cho bài toán thay vì chỉ có một đáp án đơn tốt nhất.

Để đánh giá sự phù hợp của lời giải thu được từ mô hình, một hàm thích nghi (fitness function) xem xét đến yếu tố thời gian và chi phí sẽ được áp dụng cho bài toán tối ưu đa mục tiêu TCO.

Phương pháp được sử dụng có tên gọi là *phương pháp trọng số thích ứng sửa đổi* (Modified Adaptive Weight Approach – MAWA), được Zheng và các cộng sự (2004) phát triển từ *phương pháp trọng số thích ứng* (Adaptive Weight Approach – AWA) đề xuất bởi Gen và Cheng (2000), và được sử dụng trong việc áp dụng thuật giải di truyền cho bài toán TCO.

Cụ thể, trong phương pháp MAWA, các trọng số thích ứng được tính toán theo bốn điều kiện sau:

- (1) Nếu $Z_t^{\max} \neq Z_t^{\min}$ và $Z_c^{\max} \neq Z_c^{\min}$ thì :

$$v_c = \frac{Z_c^{\min}}{Z_c^{\max} - Z_c^{\min}} \quad (7)$$

$$v_t = \frac{Z_t^{\min}}{Z_t^{\max} - Z_t^{\min}} \quad (8)$$

$$v = v_t + v_c \quad (9)$$

$$w_t = \frac{v_t}{v} \quad (10)$$

$$w_c = \frac{v_c}{v} \quad (11)$$

- (2) Nếu $Z_t^{\max} = Z_t^{\min}$ và $Z_c^{\max} = Z_c^{\min}$ thì :

$$w_t = w_c = 0.5 \quad (12)$$

- (3) Nếu $Z_t^{\max} = Z_t^{\min}$ và $Z_c^{\max} \neq Z_c^{\min}$ thì :

$$w_t = 0.9 \quad (13)$$

$$w_c = 0.1 \quad (14)$$

- (4) Nếu $Z_t^{\max} \neq Z_t^{\min}$ và $Z_c^{\max} = Z_c^{\min}$ thì :

$$w_t = 0.1 \quad (15)$$

$$w_c = 0.9 \quad (16)$$

Trong đó :

▪ $Z_t^{\max}$: giá trị cực đại theo mục tiêu thời gian trong tập hợp các đáp án Pareto thu được từ vòng lặp của thuật toán ACO.

▪ $Z_t^{\min}$: giá trị cực tiểu theo mục tiêu thời gian trong tập hợp các đáp án Pareto thu được từ vòng lặp của thuật toán ACO.

▪ $Z_c^{\max}$: giá trị cực đại theo mục tiêu chi phí trong tập hợp các đáp án Pareto thu được từ vòng lặp của thuật toán ACO.

▪ $Z_c^{\min}$: giá trị cực tiểu theo mục tiêu chi phí trong tập hợp các đáp án Pareto thu được từ vòng lặp của thuật toán ACO.

▪ V_t : giá trị theo tiêu chuẩn về thời gian

▪ V_c : giá trị theo tiêu chuẩn về chi phí

▪ V : giá trị cho dự án

▪ w_c : trọng số thích ứng theo tiêu chuẩn về chi phí

▪ w_t : trọng số thích ứng theo tiêu chuẩn về thời gian

Sau khi tính được các trọng số w_t , w_c tính hàm kết hợp thời gian & chi phí theo công thức :

$$f(x) = w_t \times \frac{z_t^{\max} - z_t + \gamma}{z_t^{\max} - z_t^{\min} + \gamma} + w_c \times \frac{z_c^{\max} - z_c + \gamma}{z_c^{\max} - z_c^{\min} + \gamma} \quad (17)$$

Trong đó :

▪ γ : hằng số dương ngẫu nhiên nằm trong khoảng $[0,1]$.

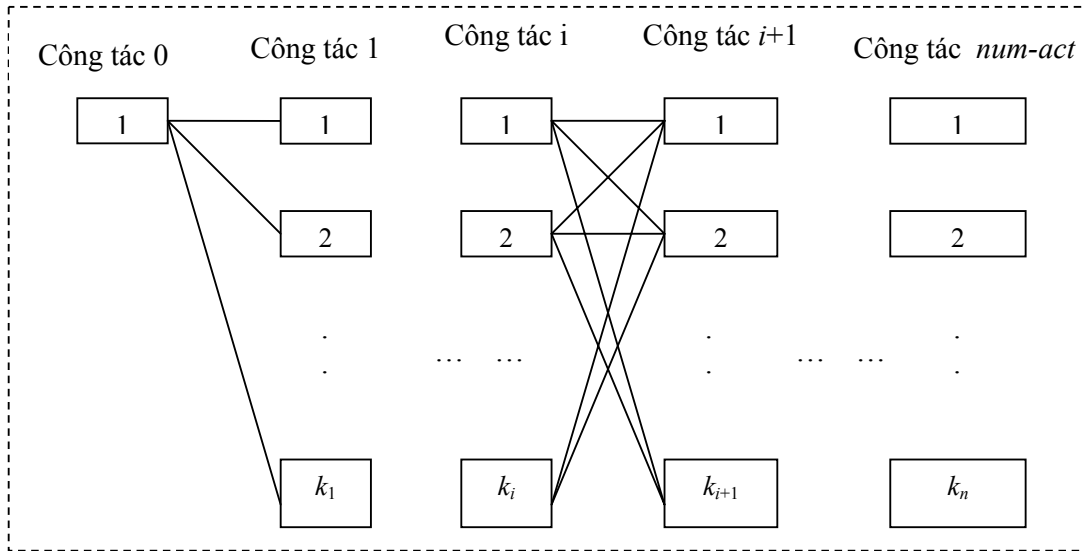
4. MÔ HÌNH ACO CHO BÀI TOÁN TCO

4.1 Mô tả bài toán

Bài toán tối ưu thời gian chi phí TCO (time-cost optimization) là một bài toán tối ưu đa mục tiêu, trong đó các đáp án là không duy nhất. Bài toán TCO tập trung chủ yếu vào việc lựa chọn những phương án thích hợp cho từng công tác nhằm đạt được mục tiêu về thời gian và mục tiêu về chi phí cho dự án.

Việc áp dụng thuật toán ACO để giải quyết bài toán TCO có thể được minh họa như sau : Đầu tiên, ta chuyển đổi bài toán TCO trở thành bài toán TSP. Sau đó, kết hợp hai mục tiêu thời gian và chi phí thành một mục tiêu nhờ vào phương pháp trọng số thích ứng sửa đổi MAWA. Cuối cùng, tìm kiếm lời giải tối ưu dựa trên thuật toán ACO.

Sự biểu diễn bài toán TCO dưới dạng TSP được mô tả trong hình (1)



Hình 1. Biểu diễn bài toán TCO dưới dạng TSP

Mỗi nút trong hình (1) biểu thị một phương án lựa chọn để thực hiện công tác. Ví dụ, nút thứ j trên cột i ($i=0,1,2,\dots$) cho biết rằng công tác i thực hiện theo phương án lựa chọn j . Cột 0 là một công tác ảo đại diện cho điểm bắt đầu của dự án. Các cạnh trên hình (1) được mô tả bởi một ma trận với 3 yếu tố, ví dụ (i, j_1, j_2) miêu tả công tác thứ i thực hiện theo lựa chọn j_1 , trong khi công tác $i+1$ thực hiện theo lựa chọn j_2 . Mỗi đường đi từ cột 0 đến cột $num-act$ trình bày một phương án thực hiện của dự án. Trên thực tế, việc giải quyết bài toán TCO là tập trung tìm kiếm một đường đi có thể làm cho cực tiểu tổng thời gian cũng như tổng chi phí của dự án.

Tổng thời gian và tổng chi phí của dự án có thể được tính toán lần lượt theo các công thức (18) và (19) sau đây :

$$T = \max \left[\sum_{i \in L_k} t_i^{(k)} x_i^{(k)} \right] \quad (18)$$

Trong đó :

- $t_i^{(k)}$: thời gian thực hiện công tác thứ i khi thực hiện theo lựa chọn thứ k .
- $x_i^{(k)}$: biến số của công tác thứ i khi thực hiện theo lựa chọn thứ k . Nếu $x_i^{(k)} = 1$ thì công tác i thực hiện theo lựa chọn thứ k ; và ngược lại nếu $x_i^{(k)} = 0$.
- Tổng của các giá trị biến số của tất cả các lựa chọn phải bằng 1.
- L_k : chuỗi công tác trên đường đi thứ k ; $L_k = \{i_{1k}, i_{2k}, \dots, i_{nk}\}$
- i_{jk} : số của công tác j trên đường đi thứ k
- L : tập hợp tất cả các đường đi của sơ đồ mạng; $L = \{L_k | k=1,2,\dots,m\}$
- m : số của các đường đi trong sơ đồ mạng.

$$C = \sum_{i \in N} dc_i^{(k)} x_i^{(k)} + T \times ic_i^{(k)} \quad (19)$$

Trong đó :

- $dc_i^{(k)}$: chi phí trực tiếp của công tác thứ i khi thực hiện theo lựa chọn thứ k , bằng với số lượng của các công tác nhân với đơn giá của chúng.

- $ic_i^{(k)}$: chi phí gián tiếp của công tác thứ i khi thực hiện theo lựa chọn thứ k , có thể tính toán bởi các chuyên gia bằng cách ước lượng hoặc thu được từ việc chia chi phí gián tiếp của ngân sách theo tổng thời gian của hợp đồng.

- N : tập hợp các công tác trong sơ đồ mạng.

4.2 Mô hình ACO-TCO

Mô hình ACO-TCO được mô tả gồm các bước chính như sau :

❖ Bước 1 : Khởi tạo các đáp án ban đầu

Trước tiên, tất cả các con kiến nhân tạo được đặt ở nút khởi đầu. Tiếp theo, tạo ra một cách ngẫu nhiên một đường đi từ nút khởi đầu đến nút kết thúc cho mỗi con kiến. Điều này có nghĩa là mỗi con kiến sẽ chọn lựa một cách ngẫu nhiên một phương án thực hiện cho mỗi công tác để tạo ra một đáp án khả thi cho bài toán TCO.

❖ Bước 2 : Tính toán tổng thời gian và chi phí của dự án

Tính toán tổng thời gian hoàn thành và tổng chi phí dự án cho mỗi đường đi được tạo ra bởi mỗi con kiến theo các công thức (18) và (19).

❖ Bước 3 : Thiết lập vùng đáp án (solution pool) và tìm các đáp án tối ưu Pareto, đặt tên là E

Mục đích của việc thiết lập vùng đáp án là làm giảm việc tính toán lặp lại một cách không cần thiết trong suốt quá trình chạy thuật toán. Khi tạo ra một đáp án mới, trước tiên sẽ tìm kiếm trong vùng đáp án. Nếu đáp án này đã xuất hiện trong vùng đáp án, thì loại bỏ nó, nếu không thì tính toán giá trị đó theo các công thức (18) và (19). Phù hợp với định nghĩa các đáp án tối ưu của Pareto, xóa đi các đáp án không trội từ vùng đáp án, phần còn lại sẽ tạo thành các đáp án tối ưu Pareto E .

❖ Bước 4 : Phân phối các trọng số cho mục tiêu thời gian và chi phí

Tìm các giá trị $Z_t^{\max}, Z_t^{\min}, Z_c^{\max}, Z_c^{\min}$ trong E , sau đó phân phối các trọng số theo mục tiêu thời gian và chi phí dựa vào các công thức từ (7) đến (16).

❖ Bước 5 : Tính toán giá trị kết hợp của mục tiêu thời gian và chi phí

Trong quá trình chuyển đổi hai mục tiêu thời gian và chi phí thành đơn mục tiêu dưới dạng trọng số, ta sẽ thu được giá trị kết hợp của mục tiêu thời gian và chi phí, giá trị này được sử dụng để sửa đổi cường độ mùi trên đường đi. Giá trị kết hợp của mục tiêu thời gian và chi phí được tính theo công thức (17).

❖ Bước 6 : Tính toán giá trị cập nhật của vết mùi trên mỗi đường đi sau một vòng lặp

Sau mỗi vòng lặp, giá trị cập nhật của vết mùi trên mỗi cạnh (i, j_1, j_2) được tính toán theo công thức sau :

$$\Delta \tau_{i,j_1,j_2} = \sum_{k=1}^{num_ant} \Delta \tau_{i,j_1,j_2}^k \quad (20)$$

Trong đó :

- num_ant : tổng số lượng kiến

▪ $\Delta \tau_{i,j_1,j_2}$: giá trị cập nhật của vết mùi trên cạnh (i,j_1,j_2) sau một vòng lặp

▪ $\Delta \tau_{i,j_1,j_2}^k$: giá trị cập nhật của vết mùi mà con kiến thứ k để lại trên cạnh (i,j_1,j_2) , được xác định như sau :

$$\Delta \tau_{i,j_1,j_2}^k = \begin{cases} Q & \text{Nếu con kiến } k \text{ đi qua cạnh } (i,j_1,j_2) \\ f(k) & \text{Ngược lại} \\ 0 & \end{cases} \quad (21)$$

▪ Q : là một hằng số, đặc trưng cho lượng mùi mà một con kiến để lại trên đường đi.

▪ $f(k)$: giá trị kết hợp của mục tiêu thời gian và chi phí của đáp án thứ k , thu được từ công thức (17)

❖ **Bước 7 : Cập nhật vết mùi trên mỗi cạnh**

$$\left\{ \begin{array}{l} p_{i,j_1,j_2}^k = \frac{[\tau_{i,j_1,j_2}]^\alpha \cdot [\eta_{i,j_1,j_2}]^\beta}{\sum_{u \in J_k(i)} [\tau_{i,j_1,u}]^\alpha \cdot [\eta_{i,j_1,u}]^\beta}, \text{ nếu } j \in J_k(i) \\ \text{Ngược lại, } p_{i,j_1,j_2}^k = 0 \end{array} \right. \quad (23)$$

Trong đó :

▪ p_{i,j_1,j_2}^k : xác suất để con kiến k lựa chọn cạnh (i,j_1,j_2) để đi

▪ α : thông số điều chỉnh ảnh hưởng của vết mùi $\Delta \tau_{i,j_1,j_2}$

Cuối mỗi vòng lặp, cường độ của vết mùi trên mỗi cạnh được cập nhật lại theo quy tắc sau:

$$\Delta \tau_{i,j_1,j_2}(nc+1) = \rho \cdot \tau_{i,j_1,j_2}(nc) + \Delta \tau_{i,j_1,j_2} \quad (22)$$

Trong đó :

▪ $\Delta \tau_{i,j_1,j_2}(nc)$: vết mùi trên cạnh (i,j_1,j_2) sau vòng lặp nc

▪ $\Delta \tau_{i,j_1,j_2}(nc+1)$: vết mùi trên cạnh (i,j_1,j_2) sau vòng lặp $nc+1$

▪ $\rho \in [0,1]$: là một hằng số, đặc trưng cho tỷ lệ tồn tại của vết mùi trước đó ; như vậy $1-\rho$ đặc trưng cho sự bay hơi của vết mùi.

❖ **Bước 8 : Tính toán xác suất lựa chọn đường đi trên mỗi cạnh của các con kiến**

Kiến lựa chọn đường đi dựa trên cường độ mùi và tầm nhìn của mỗi cạnh. Do đó, xác suất lựa chọn cho mỗi cạnh được tính theo công thức sau :

▪ β : thông số điều chỉnh ảnh hưởng của

η_{i,j_1,j_2}

▪ $J_k(i)$: tập hợp các nút mà con kiến k ở nút i chưa đi qua

▪ τ_{i,j_1,j_2} : nồng độ của vết mùi trên cạnh (i,j_1,j_2)

▪ η_{i,j_1,j_2} : thông tin heuristic (hay còn gọi là tầm nhìn) giúp đánh giá chính xác sự lựa chọn của con kiến khi quyết định đi trên cạnh

(i,j_1,j_2) , tương trưng cho thông tin cục bộ được xem xét trong quá trình ; được xác định theo công thức :

$$\eta_{i,j_1,j_2} = w_c \times \frac{dc_{i+1}^{\max} - dc_{i+1}^{(k)} + r}{dc_{i+1}^{\max} - dc_{i+1}^{\min} + r} + w_t \times \frac{t_{i+1}^{\max} - t_{i+1}^{(k)} + r}{t_{i+1}^{\max} - t_{i+1}^{\min} + r} \quad (24)$$

Với :

▪ $dc_{i+1}^{\max}$: giá trị cực đại chi phí trực tiếp của công tác $i+1$ theo những lựa chọn khác nhau.

▪ $dc_{i+1}^{\min}$: giá trị cực tiểu chi phí trực tiếp của công tác $i+1$ theo những lựa chọn khác nhau.

▪ $t_{i+1}^{\max}$: giá trị cực đại về thời gian thực hiện công tác $i+1$ theo những lựa chọn khác nhau.

▪ $t_{i+1}^{\min}$: giá trị cực tiểu về thời gian thực hiện công tác $i+1$ theo những lựa chọn khác nhau.

▪ $dc_{i+1}^{(k)}$: chi phí trực tiếp của công tác $i+1$ khi thực hiện theo lựa chọn thứ k

▪ $t_{i+1}^{(k)}$: thời gian thực hiện của công tác $i+1$ khi thực hiện theo lựa chọn thứ k

❖ **Bước 9 : Lựa chọn đường đi cho mỗi con kiến**

Để lựa chọn thực hiện một công tác, con kiến sẽ sử dụng thông tin heuristic biểu thị bởi η_{i,j_1,j_2} cũng như là thông tin về vết mùi biểu thị bởi $\Delta\tau_{i,j_1,j_2}$. Quy tắc lựa chọn được mô tả bởi công thức sau đây :

$$j = \left\{ \arg \max_{u \in J_k(i)} \left[(\tau_{i,j_1,u})^\alpha \times (\eta_{i,j_1,u})^\beta \right] \right\} \quad (25)$$

Trong đó :

▪ q : giá trị được lựa chọn một cách ngẫu nhiên với một xác suất không thay đổi trong khoảng $[0,1]$.

▪ $0 \leq q_0 \leq 1$: là một tham số cho trước.

▪ J : là một biến số ngẫu nhiên được lựa chọn theo sự phân bố xác suất cho bởi công thức (23).

❖ **Bước 10 :**

Thêm đáp án mới từ quá trình vào vùng đáp án, và cập nhật các đáp án tối ưu Pareto E . Lặp lại quá trình từ Bước 4 đến Bước 10 cho đến khi điều kiện kết thúc (phương trình (7), (16), (17), (20), (22), (23), (24), (25)) được thỏa mãn.

5. VÍ DỤ MINH HỌA

Để minh họa cho tính hiệu quả của mô hình đề xuất, một chương trình máy tính ứng dụng các bước của mô hình trên đã được thực hiện. Chương trình được viết bằng ngôn ngữ lập trình Visual Basic 6.0. Việc thực hiện mô hình nhằm cố gắng tạo ra một chương trình thân thiện và dễ sử dụng.