

TỔNG QUAN VỀ MỘT SỐ PHƯƠNG PHÁP GIẢI HỆ RÀNG BUỘC QUÁ MỨC

AN OVERVIEW OF CURRENT APPROACHES FOR SOLVING OVER-CONSTRAINED PROBLEMS

Dương Tuấn Anh

Khoa Công Nghệ Thông Tin, Đại học Bách khoa, Tp. Hồ Chí Minh, Việt nam

BẢN TÓM TẮT

Bài viết này trình bày khái quát những cách tiếp cận hiện hành để mô hình hóa và giải những hệ ràng buộc quá mức. Những bài toán này thường được thiết lập như là những bài toán tối ưu tổ hợp và có tồn tại nhiều cách hình thức hóa khác nhau để mô hình chúng. Về phương pháp giải, bài viết sẽ khảo sát cả hai cách tiếp cận: tìm kiếm có hệ thống và tìm kiếm cục bộ. Cuối cùng bài viết sẽ nói về một xu hướng mới trong phương pháp giải hệ ràng buộc quá mức: sự lai ghép giữa tìm kiếm có hệ thống và tìm kiếm cục bộ.

ABSTRACT

This paper reviews existing approaches to model and solve over-constrained problems. These problems are usually formulated as combinatorial optimization problems, and there exist different formalisms to model them. Regarding solving methods, both systematic search and local search approaches are considered. Finally, we discuss a new trend in solving approaches for over-constrained systems: hybridization of these two above-mentioned paradigms.

1. GIỚI THIỆU

Bài toán giải hệ ràng buộc (constraint satisfaction problem), viết tắt là CSP, được định nghĩa bằng một bộ ba (V, D, C) :

$V = \{V_1, \dots, V_n\}$ là tập các biến,

$D = \{D_1, \dots, D_n\}$ là tập các miền trị (domain) hữu hạn D_i mà D_i là tập các trị cho phép của V_i .

$C = \{C_1, \dots, C_r\}$ là tập các ràng buộc. Mỗi ràng buộc C_i trên tập biến $var(C_i)$ đặc tả quan hệ $rel(C_i)$ của các tổ hợp trị cho các biến trong $var(C_i)$; $rel(C_i)$ là một tập con của tích đề các của các miền trị của các biến trong $var(C_i)$.

Một lời giải của một CSP là một phép gán trị đầy đủ vào các biến mà thỏa mãn mọi ràng buộc. Bài toán CSP còn có thể xem như một bài toán quyết định, mà mục tiêu là quyết định xem có tồn tại một lời giải hay không bằng cách tạo ra một lời giải như vậy. Một cách giải CSP tiêu biểu là một thủ tục quay lui (backtracking) tìm kiếm trên cây (xem Kumar, 1992 [14]).

Nếu tất cả các ràng buộc trong một CSP đều là bắt buộc, thì điều này có khi sẽ khiến cho bài toán trở nên không thể giải được. Tình hình này là nguyên nhân cho việc mở rộng lược đồ CSP để bao gồm các hệ ràng buộc quá mức (over-constrained systems). Một CSP

ràng buộc quá mức, viết tắt OCSP, là một CSP mà mọi phép gán trị đầy đủ đều vi phạm một vài ràng buộc nào đó.

Một lời giải cho một OCSP là một phép gán trị vào biến mà thỏa tốt nhất tập ràng buộc.

Có một số mô hình cho hệ ràng buộc quá mức đã được đề xuất, có thể kể như: 1) hệ ràng buộc riêng phần (partial constraint satisfaction problem), 2) hệ phân cấp ràng buộc (constraint hierarchy) và 3) hệ ràng buộc có trọng số.

Giải hệ ràng buộc quá mức là một công việc tối ưu hóa. Đó là một bài toán NP-đầy đủ, thường khó hơn giải một bài toán CSP cổ điển. Nếu những vi phạm ràng buộc có thể thu gộp thành một hàm đơn, giải OCSP có thể coi như giải một bài toán tối ưu hóa đơn mục tiêu. Ngược lại, giải OCSP trở thành việc giải một bài toán tối ưu hóa đa mục tiêu.

Có hai cách tiếp cận chính để giải hệ ràng buộc quá mức: tìm kiếm có hệ thống và tìm kiếm cục bộ. Các phương pháp tìm kiếm có hệ thống (systematic search) thực hiện một sự duyệt có hệ thống trong không gian trạng thái, viếng thăm mọi trạng thái mà có thể chứa đựng một lời giải. Các phương pháp này đem lại lời giải tối ưu cho bài toán. Tìm kiếm cục bộ (local search) thực hiện việc tìm kiếm trên một vài không gian con của toàn bộ không gian trạng thái, tìm thấy một lời giải tốt trong một khoảng thời gian để chấp nhận. Mặc dù những phương pháp này chỉ đem lại lời giải cận tối ưu (near-optimal solution), nhưng ngày càng được ưa chuộng vì chúng giải được những bài toán có quy mô lớn với tập ràng buộc khá phức tạp với một mức độ hữu hiệu đáng kể về thời gian tính toán.

Bài viết này được tổ chức như sau. Phần 2 trình bày những mô hình dùng để thiết lập các bài toán giải hệ ràng buộc quá mức. Trong phần 3, chúng tôi điếm qua các phương pháp giải chính để giải các hệ ràng buộc quá mức. Kể cả những giải thuật lai ghép. Phần 4 đúc kết

và vạch ra những hướng có triển vọng cho những nghiên cứu trong tương lai.

2. MÔ HÌNH HÓA BÀI TOÁN GIẢI HỆ RÀNG BUỘC QUÁ MỨC

2.1 Hệ ràng buộc riêng phần (Partial constraint satisfaction problem)

Giải hệ ràng buộc riêng phần ([9]) là tìm tổ hợp trị gán vào tập biến mà thỏa một tập con của tập ràng buộc C. Chúng ta sẵn lòng “làm yếu” một vài ràng buộc để có thể tìm ra một vài tổ hợp trị khả thi. Giải hệ ràng buộc riêng phần nảy sinh ra trong tình huống phải giải một hệ ràng buộc quá mức.

Định nghĩa: Bài toán giải hệ ràng buộc riêng phần (PCSP) bao gồm một bộ ba (V, C, ω) , V là tập biến, C là tập các ràng buộc và hàm $\omega: C \rightarrow \mathcal{R}$ ánh xạ các ràng buộc vào các trọng số. Trọng số của một ràng buộc biểu thị tầm quan trọng của nó. Người ta có thể mô tả những ràng buộc cứng mà nhất thiết phải thỏa mãn cùng với những ràng buộc mềm, là những ràng buộc nên thỏa càng nhiều càng tốt. Một ràng buộc cứng được cho trọng số lớn vô hạn.

Lời giải của một PCSP là một phép gán trị vào biến sao cho hoặc là

- tổng số các ràng buộc bị vi phạm là tối thiểu

hoặc

- tổng trọng số các ràng buộc bị vi phạm là tối thiểu.

Trong trường hợp chúng ta cố gắng tìm một lời giải thỏa được càng nhiều ràng buộc càng tốt (trường hợp 1), thì loại PCSP này còn được gọi là bài toán giải hệ ràng buộc tối đa (maximal CSP).

2.2. Hệ ràng buộc có trọng số (weighted constraint satisfaction problem)

Trong bài toán giải hệ ràng buộc có trọng số (WCSP), mỗi ràng buộc được gán một trọng số hay chi phí. Chi phí của một phép gán trị vào biến là tổng chi phí của các ràng buộc bị vi phạm. Lời giải của một WCSP là một sự gán trị vào biến sao cho chi phí của phép gán trị này là tối thiểu.

Như vậy, hệ ràng buộc có trọng số (WCSP) chỉ là một trường hợp riêng (trường hợp thứ hai) của hệ ràng buộc riêng phần (PCSP)

2.3. Hệ phân cấp ràng buộc (constraint hierarchy)

Trong hệ phân cấp ràng buộc ([2]), tập ràng buộc C được tổ chức thành một danh sách C_H có dạng $\langle C_0, C_1, \dots, C_n \rangle$, với mỗi C_i , $0 \leq i \leq n$, là một tập các ràng buộc ở mức i .

Mỗi mức mô tả tầm quan trọng của tập ràng buộc C_i đó. Các ràng buộc ở mức 0, thuộc tập C_0 được gọi là các ràng buộc bắt buộc (ràng buộc cứng) là các ràng buộc nhất thiết phải thỏa mãn. Các ràng buộc trong các tập C_1, C_2, \dots, C_n chỉ những ràng buộc nhiệm ý (ràng buộc mềm). Các ràng buộc mềm này bao gồm từ mức mạnh nhất C_1 cho đến mức yếu nhất C_n .

Gọi θ là một phép gán trị vào biến, là một tổ hợp trị mà mỗi trị lấy từ miền trị của các biến tương ứng. Tập các lời giải S của một hệ phân cấp ràng buộc bao gồm những phép gán trị θ . Mỗi phép gán trị thuộc tập S trước tiên phải thỏa tất cả các ràng buộc bắt buộc (tức tập C_0). Ngoài ra, ta muốn mỗi phép gán trị θ đó phải thỏa các ràng buộc nhiệm ý càng nhiều càng tốt và có tầm quan trọng tương đối của chúng. Trong định nghĩa sau, $c\theta$ biểu thị kết quả logic của việc kiểm tra phép gán trị θ đối với ràng buộc c , và ta bảo $c\theta$ đúng nếu θ thỏa ràng buộc c .

$$S_0 = \{ \theta \mid \forall c \in C_0, c\theta \text{ đúng} \}$$

$$S = \{ \theta \mid \theta \in S_0 \wedge \forall \sigma \in S_0 \neg \text{better}(\sigma, \theta, H) \}$$

Trong định nghĩa tập lời giải S , ta cần một vị từ so sánh *better*. Có nhiều cách định nghĩa vị từ so sánh *better*. Trong trường hợp thông

dụng nhất, vị từ *locally-predicate-better* thường được dùng. Vị từ này xem xét từng ràng buộc một trong hệ phân cấp ràng buộc và kiểm xem ràng buộc đó có thỏa hay không. Một phép gán trị θ là *locally-predicate-better* hơn phép gán trị σ nếu θ thỏa được tất cả các mức ràng buộc mà σ thỏa cho đến một mức $k-1$ nào đó và đặc biệt θ thỏa được nhiều ràng buộc hơn σ tại mức k .

Có thể nói mô hình hệ phân cấp ràng buộc diễn tả các hệ ràng buộc quá mức một cách tự nhiên và tinh xảo hơn hệ ràng buộc có trọng số. Ngoài ra, bằng một phương án gán trọng số vào ràng buộc hợp lý, tầm quan trọng tương đối giữa các mức, thì ta có thể chuyển một hệ phân cấp ràng buộc thành một hệ ràng buộc có trọng số.

Có tồn tại một giải thuật để giải hệ phân cấp ràng buộc trên miền trị hữu hạn, được đề xuất bởi Menezes et al., 1993 ([15]). Giải thuật này là một giải thuật dạng tăng dần (incremental) và dựa vào những kỹ thuật của lập trình ràng buộc. Giải thuật này thuộc họ phương pháp tìm kiếm có hệ thống

3. CÁC PHƯƠNG PHÁP GIẢI

3.1 Các phương pháp tìm kiếm có hệ thống (systematic search)

Giải thuật Nhánh-và-Cận

Tất cả các mô hình của OCSP được trình bày ở phần 2 đều có thể giải bằng những phương pháp dựa vào nhánh-và-cận tìm theo chiều sâu trước (depth-first branch-and-bound), một giải thuật tối ưu hóa để giải bài toán tối ưu tổ hợp trên miền trị hữu hạn.

Nhánh-và-cận (BB) thực hiện sự duyệt theo chiều sâu trước (depth-first traversal) trên cây tìm kiếm được định nghĩa bởi bài toán, mà trong đó các nút nội diễn tả những phép gán trị chưa đầy đủ và các nút lá diễn tả những phép gán trị đầy đủ. Các biến đã được gán trị được

gọi là các *biến quá khứ* (P), còn các biến chưa được gán trị được gọi là các *biến tương lai* (F). *Khoảng cách* của một nút (tức một phép gán trị) là số ràng buộc bị vi phạm bởi phép gán trị của nút ấy. BB cần dùng một cận cần N và một cận đủ S đối với một lời giải có thể chấp nhận được. Giá trị khởi tạo của cận S là dựa vào bài toán cụ thể. Giá trị khởi tạo của N là vô hạn. Tại mỗi nút giải thuật BB tính N như là khoảng cách của lời giải tốt nhất đạt được cho đến bây giờ. Nếu tại một nút, khoảng cách của phép gán trị chưa đầy đủ hiện hành lớn hơn hay bằng cận N thì giải thuật BB sẽ cắt tĩa các nút con thuộc nhánh đó và quay lui.

Giải thuật BB được trình bày ở hình 1. Thủ tục gồm ba tham số: P là tập các biến quá khứ, F là tập các biến tương lai, và FD là tập miền trị của các biến tương lai. Trong thủ tục, biến X_i chỉ biến hiện hành, a chỉ trị được chọn để gán vào biến hiện hành.

```

procedure BB(P, F, FD)
if (F =  $\emptyset$ ) then
    BestS  $\leftarrow$  assignment(P);
    N  $\leftarrow$  distance(BestS);
    if N  $\leq$  S then return ‘finished’
    else return ‘keep-searching’ /*
backtrack */
else
    Xi  $\leftarrow$  select-variable(F);
    while FDi  $\neq$   $\emptyset$  do
        a  $\leftarrow$  select-value(FDi);
        if (distance(P  $\cup$  {Xi, a}) < N) then
            BB(P  $\cup$  {Xi, a}, F - {Xi}, FD -
{FDi});
            FDi  $\leftarrow$  FDi - {a}
    endprocedure
    
```

Hình 1. Giải thuật nhánh-và-cận tìm theo chiều sâu trước

Các giải thuật Nhánh và Cận cải tiến

Một dạng cải tiến của giải thuật nhánh-và-cận là giải thuật *nhánh-và-cận có kiểm tra hướng tới* (BB-FC) phối hợp phương pháp

nhánh-và-cận và kỹ thuật kiểm tra hướng tới (forward checking), đề xuất bởi Freuder và Wallace, 1992 ([9]). Mỗi khi một trị a được gán vào một biến hiện hành v nào đó, thì giải thuật sẽ kiểm tra tất cả các biến tương lai mà có dính líu một ràng buộc với v và loại ra khỏi những miền trị của những biến này bất cứ trị nào không tương thích với trị a . Sự thu hẹp một miền trị của một biến tương lai thành tập rỗng sẽ phát sinh một tình huống bị bế tắc (phải chuyển sang trị khác hoặc quay lui).

Trong khuôn khổ giải thuật BC-FC, có những cách khác nhau để tính mức độ vi phạm ràng buộc của lời giải chưa đầy đủ hiện hành, mỗi cách tính khác nhau đưa lại những giải thuật BB-FC khác nhau. Những giải thuật đó được tổng kết trong bài báo của Meseguer et al. ([17]).

Cũng giống như việc giải các hệ ràng buộc (CSP) thông thường, các giải thuật BB hay BB-FC có thể sử dụng thêm các heuristic sắp thứ tự biến và sắp thứ tự trị để cải tiến tính hữu hiệu của giải thuật ([9],[17]).

3.2 Các phương pháp tìm kiếm cục bộ

Những bài toán tối ưu tổ hợp trong thực tế thường có hai đặc điểm mà làm cho chúng trở nên quá khó giải: 1) qui mô bài toán quá lớn và 2) bài toán *không thuần nhất*, tức là bài toán có một tập hợp ràng buộc gồm nhiều dạng hỗn tạp của các ràng buộc phụ trợ (side constraint). Trong hầu hết các trường hợp thực tế như vậy, các phương pháp chính xác và các phương pháp tìm kiếm có hệ thống đã không thể nào giải được trong khi đó các phương pháp tìm kiếm cục bộ đã tỏ ra hữu hiệu trong việc tìm ra được những lời giải rất tốt cho bài toán.

Tìm kiếm cục bộ dựa vào một ý tưởng tổng quát và đơn giản. Gọi P là một bài toán tối ưu tổ hợp cần giải, và s là lời giải hiện hành giả sử là một lời giải khả thi của P , và có hàm chi phí $f(s)$. *Miền lân cận* $N(s)$ được định nghĩa cho s , là tập những lời giải láng giềng khả thi s' của s sao cho từ s ta có thể đạt tới s' nhờ vào

một *bước chuyển* m . Bước chuyển có tác dụng biến đổi s thành ra một lời giải láng giềng. Thao tác biến đổi này được lặp cho đến khi hội tụ về một lời giải tốt. Lời giải này là *lời giải cận tối ưu*, mà trong một số bài toán thực tế, không sai biệt gì nhiều với lời giải tối ưu.

3.2.1 Xung đột tối thiểu (Min-conflict)

Giải thuật *xung đột tối thiểu*, viết tắt là MC (Minton et al., 1990 [18]; Minton et al., 1992 [19]) đã được dùng khá phổ biến để giải hệ ràng buộc quá mức. Giải thuật MC chọn ngẫu nhiên một biến nào đó dính líu đến một ràng buộc bị vi phạm và rồi chọn một trị từ miền trị của biến này sao cho tối thiểu hoá số lượng những vi phạm ràng buộc có thể xảy ra.

Vì giải thuật MC thuần túy có thể không thoát ra được điểm tối ưu cục bộ, giải thuật thường kết hợp với một chiến lược *bước ra ngẫu nhiên* (random walk). Với một biến nào đó được chọn, chiến lược *bước ra ngẫu nhiên* lấy ngẫu nhiên một trị từ miền trị của biến này với xác suất p , và áp dụng theo giải thuật MC với xác suất $1-p$. Giá trị của thông số p có ảnh hưởng lên hiệu quả của giải thuật. Giải thuật này được gọi là MCRW.

3.2.2 Giải thuật mô phỏng luyện kim (simulated annealing)

Mô phỏng luyện kim(SA) (Kirkpatrick et al., 1983 [13]) là một kỹ thuật tìm kiếm ngẫu nhiên (stochastic search) mà tỏ ra rất hữu hiệu cho những bài toán tối ưu hóa qui mô lớn. Trong kỹ thuật này, nhiệt độ là biến được khởi tạo ở một giá trị cao và dần dần giảm dần xuống trong quá trình tìm kiếm. Tại những trị nhiệt độ cao, các bước chuyển được chấp nhận một cách ngẫu nhiên bất luận chúng là bước chuyển có cải thiện hàm chi phí của lời giải hay không. Khi nhiệt độ được giảm xuống, xác suất để chấp nhận một lời giải có cải thiện sẽ tăng lên và xác suất để chấp nhận một lời giải không cải thiện sẽ giảm xuống. Có một số cách thức giảm nhiệt độ dần xuống được dùng

trong một giải thuật SA, được gọi là *lịch biểu làm nguội* (cooling schedule).

3.2.3 Tìm kiếm Tabu (Tabu search)

Tìm kiếm Tabu được đề xuất bởi Glover năm 1986 ([10]). Phương pháp dò tìm trong không gian lời giải bằng cách di chuyển từ một lời giải s tại lượt lặp t về một lời giải tốt nhất s' trong tập con N^* của miền lân cận $N(s)$. Vì s' không nhất thiết cải thiện chi phí của s , một cơ chế được đặt ra để ngăn chặn quá trình khởi lập vòng trên một chuỗi các lời giải. Một cách để tránh sự lặp vòng là cấm quá trình tìm kiếm quay về những lời giải đã gặp rồi, nhưng làm như vậy đòi hỏi phải lưu trữ khá nhiều thông tin. Thay vì làm thế, chỉ một vài thuộc tính của những lời giải đã gặp sẽ được lưu trong *danh sách tabu* (tabu list) và bất kỳ lời giải nào sở hữu những thuộc tính này sẽ không được xét đến trong θ lần lặp. Cơ chế này thường được gọi là *bộ nhớ ngắn hạn* và θ được gọi là *kỳ hạn tabu*.

Tìm kiếm tabu được phát triển thành nhiều dạng cải tiến như tìm kiếm tabu thích nghi (reactive tabu search) ([1]) và tìm kiếm tabu với hai danh sách tabu: *bộ nhớ ngắn hạn* và *bộ nhớ dài hạn* ([32]).

3.2.3 Giải thuật di truyền (genetic algorithm)

Giải thuật di truyền (GA) (Goldberg, 1989 [11]) đã tỏ ra khá thành công trong một số những áp dụng. GA mượn ý tưởng trong quá trình tiến hóa của sinh vật. Ý tưởng chính của giải thuật là duy trì một quần thể các lời giải ứng viên. Các lời giải ứng viên này sẽ được cho cơ hội riêng lẻ để sản sinh ra con cái tùy thuộc vào *độ thích nghi* (fitness) của chúng. Độ thích nghi được đo bằng một hàm mục tiêu. Giải thuật GA đã được áp dụng vào việc giải hệ ràng buộc ([5],[6]).

Việc dùng giải thuật GA vào các bài toán tối ưu hóa có ràng buộc làm phát sinh nhiều vấn đề mà các nhà nghiên cứu phải quan tâm giải quyết. Một trong những vấn đề quan trọng

là làm thế nào để đưa các ràng buộc vào các hàm thích nghi (fitness function) để điều khiển quá trình tìm kiếm một cách đúng đắn. Để đưa các ràng buộc vào, có một cách tiếp cận là dùng các hàm phạt (penalty function).

Ngoài bốn giải thuật tìm kiếm cục bộ nổi tiếng nêu trên, một số phương pháp tìm kiếm cục bộ khác cũng được phát triển và sử dụng, có thể kể như WalkSAT ([28]), tìm kiếm cục bộ có hướng dẫn (guided local search) ([31]), tìm kiếm cục bộ lặp (iterated local search) ([30]) và tìm kiếm với miền lân cận thay đổi (variable neighborhood search) ([31]).

Sự thành công của bất kỳ giải thuật tìm kiếm cục bộ nào nêu trên cũng tùy thuộc vào các đặc điểm thi công, tức là tùy thuộc vào các tham số kỹ thuật đặc thù mà người sử dụng phải xác định khi áp dụng giải thuật tìm kiếm cục bộ đó vào bài toán cụ thể. Quá trình thực nghiệm để xác định các thông số kỹ thuật của một giải thuật tìm kiếm cục bộ nào đó khi áp dụng vào một bài toán cụ thể được gọi là quá trình điều chỉnh thông số (parameter tuning).

3.3 Xu hướng mới: Các phương pháp giải kết hợp

Nhiều bài toán giải hệ ràng buộc quá mức trong thực tế có thể được giải một cách hữu hiệu hơn nữa bằng cách kết hợp hai cách tiếp cận: tìm kiếm có hệ thống và tìm kiếm cục bộ. Sự kết hợp này dựa trên cơ sở phát huy thế mạnh của cả hai phương pháp vào trong một giải thuật. Tìm kiếm có hệ thống, đặc biệt là các kỹ thuật của lập trình ràng buộc, mà nhánh-và-cận và lan truyền ràng buộc (constraint propagation) là những kỹ thuật tiêu biểu, có ưu thế về tính mềm dẻo theo đó việc thêm vào các ràng buộc mới là dễ dàng, không gây ảnh hưởng đến mô hình và giải thuật đã được xác lập. Tìm kiếm cục bộ lại có ưu thế lớn về tính hữu hiệu thời gian tính toán. Sự kết hợp giữa hai phương pháp này trong cùng một giải thuật hứa hẹn tạo ra được những thuật toán khá mạnh để giải lớp bài toán OCSP. Do

vậy, sự kết hợp này là một xu hướng nghiên cứu đang phát triển mạnh mẽ, tạo ra những đề tài nghiên cứu mở.

3.3.1 Kết hợp giữa tìm kiếm có hệ thống và tìm kiếm cục bộ

Một khuynh hướng kết hợp giữa tìm kiếm có hệ thống và tìm kiếm cục bộ nổi bật nhất, cần nói đến là sự lai ghép giữa lập trình ràng buộc (constraint programming) và tìm kiếm cục bộ ([7]). Lập trình ràng buộc ở đây bao gồm tất cả những kỹ thuật tìm kiếm có hệ thống đã dùng để giải các bài toán CSP và PCSP ([14], [9]). Lập trình ràng buộc và tìm kiếm cục bộ có thể kết hợp theo một trong hai cách sau đây:

- Cách 1 (LS + CP): Khung thức chính của giải thuật là tìm kiếm cục bộ, trong đó tại mỗi bước lặp của giải thuật, lập trình ràng buộc được dùng để thu hẹp miền lân cận, dẫn đến giảm thiểu chi phí tính toán trong việc tìm kiếm bước chuyển tốt nhất trong miền lân cận. Kiểu lai ghép này rất thích hợp cho những bài toán có miền lân cận quá lớn, không tiện áp dụng những giải thuật tìm kiếm cục bộ thuần túy. Tiêu biểu của cách kết hợp này là giải thuật được đề xuất bởi Pesant và Gendreau ([24]), giải thuật này dùng kỹ thuật nhánh và cận trong lập trình ràng buộc để khảo sát miền lân cận, ngầm sử dụng lan truyền ràng buộc và sự tủa nhánh để loại bớt một số lời giải lằng giằng và như vậy thu hẹp số lời giải phải xem xét. Các giải thuật kết hợp theo cách tương tự có thể kể là giải thuật giải bài toán định lộ trình xe (vehicle routing problem) của Shaw ([29]), giải thuật giải bài toán định lộ trình xe với khung thời gian hạn định (vehicle routing problem with time windows) của Rousseau et al. ([26]). Các giải thuật vừa nêu tỏ ra ưu việt hơn hẳn các giải thuật tìm kiếm cục bộ đơn thuần đã được sử dụng trước đó cho cùng bài toán.

- Cách 2 (CP+LS): Khung thức chính của giải thuật là một thủ tục tìm kiếm trên cây (tree search) theo tinh thần của lập trình ràng

buộc và dùng tìm kiếm cục bộ để cải thiện những lời giải chưa đầy đủ (partial solution) trong mỗi bước của quá trình tạo dựng lời giải. Tiêu biểu của cách kết hợp này là giải thuật được đề xuất bởi Schearf ([27]), giải thuật này xây dựng lời giải dần dần theo tinh thần của tìm kiếm trên cây và thực hiện tìm kiếm cục bộ lên lời giải chưa đầy đủ mỗi khi lời giải chưa đầy đủ này gặp phải điểm bế tắc (dead-end).

Bảng 1 liệt kê các giải thuật lai ghép thành công nhất giữa lập trình ràng buộc (CP) và tìm kiếm cục bộ (LS) với các bài toán thể nghiệm tương ứng. Các giải thuật lai ghép này có hai ưu điểm nổi bật là giải được những bài toán có qui mô lớn và thích ứng một cách dễ dàng khi

có yêu cầu thêm vào hay sửa đổi các ràng buộc phụ trợ (side constraint).

3.3.2 Kết hợp giữa hai phương pháp tìm kiếm cục bộ

Một xu hướng chính trong sự lai ghép giữa hai phương pháp tìm kiếm cục bộ với nhau là lai ghép giải thuật di truyền và tìm kiếm cục bộ. Sự lai ghép này nhằm mục đích nâng cao tính hữu hiệu về thời gian tính toán cũng như chất lượng lời giải của giải thuật di truyền. Sự lai ghép phổ biến là vận dụng một phương pháp tìm kiếm cục bộ như là một toán tử (operator) trong giải thuật di truyền. Preux et al. ([25]) gọi cách làm này là *phép lai ghép đồng bộ song song* (parallel synchronous hybridization).

Bảng 1 Các trường hợp lai ghép giữa lập trình ràng buộc và tìm kiếm cục bộ

Cách kết hợp	Tác giả	Bài toán thể nghiệm
Cách 1: LS + CP	Shaw ([29]) Pesant et al. ([24]) Rousseau et al. ([26])	vehicle routing problem (VRP) TSPTW VRPTW
Cách 2: CP + LS	Jussien et al. ([12]) Schearf ([27]) Chatzikokolakis et al. ([4]) Muller et al. ([22])	Open shop scheduling Course scheduling Open shop scheduling Course scheduling

Viết tắt: CP: constraint programming, TSPTW: traveling salesman problem with time windows, VRPTW: vehicle routing problem with time windows.

Moscalo et al. ([21]) vận dụng giải thuật leo đồi như là toán tử đột biến đã đem lại một giải thuật di truyền lai được gọi là *giải thuật memetic*, tức là một giải thuật di truyền mà sử dụng tìm kiếm cục bộ khá nhiều. Burke et al. ([3]) đã áp dụng khá thành công giải thuật memetic vào bài toán xếp lịch thi trường đại học. Ombuki et al. ([23]) vận dụng tìm kiếm cục bộ như là toán tử đột biến đã đem lại một giải thuật di truyền lai được gọi là giải thuật *LSGA*, áp dụng rất sáng tạo vào việc giải bài toán job-shop-scheduling. Merz et al. ([16]) đưa tìm kiếm cục bộ vào cả toán tử lai ghép lẫn toán tử đột biến đã đem lại một giải thuật di

truyền lai được gọi là giải thuật *GLS*, áp dụng vào việc giải bài toán người bán hàng du lịch (traveling salesman problem-TSP). Độc đáo nhất là sự lai ghép giữa giải thuật di truyền và mô phỏng luyện kim (SA) do Fotakis et al. ([8]) đưa ra và thể nghiệm với bài toán tô màu đồ thị. Fotakis et al. dùng mô phỏng luyện kim để cải thiện từng cá thể sau khi được sinh ra từ phép lai ghép và từng cá thể vừa được đột biến.

Bảng 2 liệt kê các giải thuật lai ghép thành công nhất giữa giải thuật di truyền và tìm kiếm cục bộ với các bài toán thể nghiệm tương ứng. Các giải thuật lai ghép này tỏ ra có khả năng

đem lại những lời giải tối ưu cục bộ có chất lượng cao, mặc dù lý do thành công của chúng thường vẫn còn chưa được rõ ràng.

Bảng 2. Các trường hợp lai ghép giữa hai phương pháp tìm kiếm cục bộ

Cách kết hợp	Tác giả	Bài toán thử nghiệm
GA + SA	Fotakis et al. ([8])	graph coloring
GA + HC (memetic)	Moscalo et al. ([21])	TSP
GA + LS	Merz et al. ([16])	TSP
GA + LS	Ombuki et al. ([23])	Job shop scheduling

Viết tắt: LS: local search, HC: hill-climbing, TSP: traveling salesman problem

4. KẾT LUẬN

Trong bài viết này chúng tôi tổng kết ngắn gọn những phương pháp hiện hành để mô hình hóa và giải các bài toán giải hệ ràng buộc quá mức. Về các phương pháp giải, bài viết giới thiệu cả hai họ giải thuật: tìm kiếm có hệ thống và tìm kiếm cục bộ, có để ý đến những kỹ thuật đặc thù để giải các bài toán OSCP. Một họ giải thuật mới cũng được khảo sát là họ giải thuật lai ghép: lai ghép giữa lập trình ràng buộc và tìm kiếm cục bộ hoặc lai ghép giữa giải thuật di truyền và tìm kiếm cục bộ. Hai hướng lai ghép này đem lại những giải thuật mới khá mạnh để đối phó với những bài toán OSCP qui mô lớn. Tuy vậy sự lai ghép trên đòi hỏi sự hiểu biết rất sâu vào bản chất của bài toán ứng dụng cụ thể để vận dụng một cách sáng tạo thế mạnh của từng giải thuật riêng khi đưa vào lai ghép. Do đó những công trình khảo sát cơ sở lý thuyết làm nền tảng cho một sự lai ghép đúng đắn là rất cần thiết và là một hướng nghiên cứu mở khá lý thú.

THƯ MỤC THAM KHẢO

1. R. Battiti and G. Tecchioli, "The Reactive Tabu Search", *ORSA Journal on Computing*, vol. 6, no. 2, pp.838-843, 1994.
2. A. Borning, B. Freeman-Benson, M. Wilson, "Constraint Hierarchies", *Lisp and Symbolic Computation*, Vol 5, pp.223-270, 1992.
3. E.K. Burke, J.P. Newall, R.F. Weave, "A Memetic Algorithm for University Exam Timetabling", In: E. Burke & P. Ross (Eds.) *Practice and Theory of Automated Timetabling, First International Conference, Edinburgh, U.K., August/September 1995, Selected Papers, Lecture Notes in Computer Science 1153, Springer-Verlag, 1996, pp. 241-250.*
4. K. Chatzikokolakis, G. Boukeas, P. Stamatopoulos, "Construction and Repair: A Hybrid Approach to Search in CSPs", in G.A. Vouros and T. Panayiotopoulos (Eds.): *SETN 2004, LNAI 3025, Springer-Verlag, Berlin Heidelberg 2004, pp.342-351.*
5. P. Chu and J.E. Beasley, "Genetic Algorithms for the Generalized Assignment Problem", *Computers and Operations Research*, Vol. 24, 1997, pp.17-24.
6. A. E. Eiben, P.E. Raula, Z. Ruttkay, "Solving Constraint Satisfaction using Genetic Algorithms", *Proc. of 1st IEEE Conference on Evolutionary Computing*, 1994, pp.534-547.
7. F. Focacci, F. Laburthe, A. Lodi, "Local Search and Constraint Programming", *Proc. of 4th Metaheuristics International Conference*, Porto, Portugal, July 16-20, 2001.
8. D. A. Fotakis, S.D. Likothanassis, and S. K. Stefanakos, An Evolutionary Annealing Approach to Graph Coloring, in E.J.W Boers et al. (Eds.) *EvoWorkshop 2001, LNCS 2037, Springer-Verlag, Berlin Heidelberg 2001, pp.120-129.*

9. E. Freuder, R. Wallace, Partial Constraint Satisfaction, *Artificial Intelligence*, Vol 58, 1992, pp. 21-71.
10. F. Glover and M. Laguna, “*Tabu Search*”, Kluwer Academic Publishers, 1997.
11. D. E. Goldberg, “*Genetic Algorithms in Search, Optimization and Machine Learning*”, Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.
12. N. Jussien, O. Lhomme, “Local Search with Constraint Propagation and Conflict-based Heuristics”, *Artificial Intelligence*, 139(1): pp.21-45, 2002.
13. S. Kirkpatrick, C. Gelat, and M. Vicci, “Optimization by Simulated Annealing”, *Science*, Vol. 220, No.4598, 1983, pp. 671-80.
14. V. Kumar, “Algorithms for Constraint Satisfaction Problems: A Survey”, *AI Magazine*, Spring 1992, pp.32-44.
15. F. Menezes, P. Barahona, and P. Codognet, “An Incremental Hierarchical Constraint Solver”, *PPCP’93: First Workshop on Principles and Practice of Constraint Programming*, Providence RI, 1993.
16. P. Merz, B. Freisleben, “Genetic Local Search for the TSP: New Results”, 1997.
17. P. Meseguer, N. Bouhmala, T. Bouzoubaa, M. Irgens, and M. Sanchez, “Current Approaches for Solving Over-Constrained Problems”, 2001.
18. S. Minton, M. Johnson, A. Philips, P. Laird, “Solving Large-scale Constraint Satisfaction and Scheduling Problems using a Heuristic Repair Method”, *Proc. of AAAI-90*, 1990, pp.17-24.
19. S. Minton, M. Johnson, A. Philips, P. Laird, “Minimizing Conflicts: a Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems”, *Artificial Intelligence*, Vol. 58, 1992, pp. 161-205.
20. N. Mladenovic, P. Hansen, “Variable Neighborhood Search”, *Computers in Operations Research*, vol. 24, 1997, pp.1097-1100.
21. P. Moscato, and M.G. Norman, “A Memetic Approach for the Traveling Salesman Problem Implementation of a Computational Ecology for Combinatorial Optimization on Message-Passing Systems”, *Parallel Computing and Transputer Applications*, 1992, pp. 177-186.
22. T. Muller, R. Bartak, H. Rudova, “Interactive Forward Search: Combining Local Search with Maintaining Arc Consistency and a Conflict-based Statistics”, 2003.
23. B. Ombuki, M. Ventresca, “Local Search Genetic Algorithms for the Job Shop Scheduling Problem”, Technical Report #CS-02-22, Dept. of Computer Science, Brock University, Ontario, Canada, 2002.
24. G. Pesant, M. Gendreau, “A View of Local Search in Constraint Programming”, *Proc. of Principles and Practice of Constraint Programming’ 96*, LNCS 1118, Springer-Verlag, Berlin, pp.353-366, 1996.
25. P. Preux, E-G. Talbi, “Towards Hybrid Evolutionary Algorithms”, 1997.
26. L. M. Rousseau, M. Gendreau, G. Pesant, “Using Constraint-based operators with Variable Neighborhood Search to Solve the Vehicle Routing Problem with Time Windows”, 1999.
27. A. Schaerf, “Combining Local Search and Look-ahead for Scheduling and Constraint Satisfaction Problems”, *Proc. IJCAI’97*, Nagoya, Japan, Morgan Kaufman, San Mateo, CA, 1997, pp.1254-1259.
28. B. Selman, H. Kautz, and B. Cohen, “Noise Strategies for Local Search”, *Proc. of 12th National Conference on Artificial Intelligence (AAAI’94)*, 1994, pp.337-343.
29. P. Shaw, “Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems”, *Proc. of Principle and Practice of Constraint Programming –CP98*, Pisa, Italy, October, 1998.
30. T. Stutzle, “Iterated Local Search for the Quadratic Assignment Problems”, Technical Report AIDA-99-03, FG Intellektik, Darmstadt University of Technology, Germany, 1999.
31. C. Voudouris, E. Tsang, “Guided Local Search”, Technical Report CSM-247, Dept. of Computer Science, University of Essex, 1995.
32. G. M. White, B.S. Xie, “Examination Timetables and Tabu Search with Long-

term Memory”, *Proc. of PATAT 2000*, E. Burke and W. Erben, Eds., LNCS 2079,

Springer-Verlag Berlin Heidelberg, 2001, pp.85-103.